DOCUMENT RESUME

ED 192 718 IR 008 630

TITLE	A Gateway to the Use of Computers in Education. Proceedings of the Annual Convention of the Association for Educational Data Systems (18th, St. Louis, Missouri, April 13-16, 1980).
INSTITUTION	Association for Educational Data Systems, Washington, D.C.
PUB DATE	Apr 80
NOTE	227p.; For a related document, see ED 175 446.
AVAILABLE FROM	Association for Educational Data Systems, 1201 Sixteenth Street, N.W., Washington, DC 20036 ($12.00).
EDRS PRICE	MF01 Plus Postage. PC Not Available from EDRS.
DESCRIPTORS	*Computer Assisted Instruction; *Computer Managed Instruction; *Computer Oriented Programs; Computer Science Education; Educational Media; *Educational Technology; Elementary Secondary Education; Higher Education; Laboratories; *Management Information Systems; Man Machine Systems; *Microcomputers; Online Systems
IDENTIFIERS	Computer Literacy

ABSTRACT

The 45 papers in this collection are contributed, refereed papers submitted for publication in the proceedings of the 18th Annual Association for Educational Data Systems Convention. They cover educational applications of computers in the following areas: computer assisted instruction, computer managed instruction, instructional and learning processes, computer-related curriculum, educational administration, computer resources, and data-center administration. Many papers are introduced by author abstracts and contain research methods, results, and lists of references cited. Information regarding the "vendor sessions" is not included. (CHC)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED192718

Association For Educational Data Systems

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
NATIONAL INSTITUTE OF
EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

AEDS

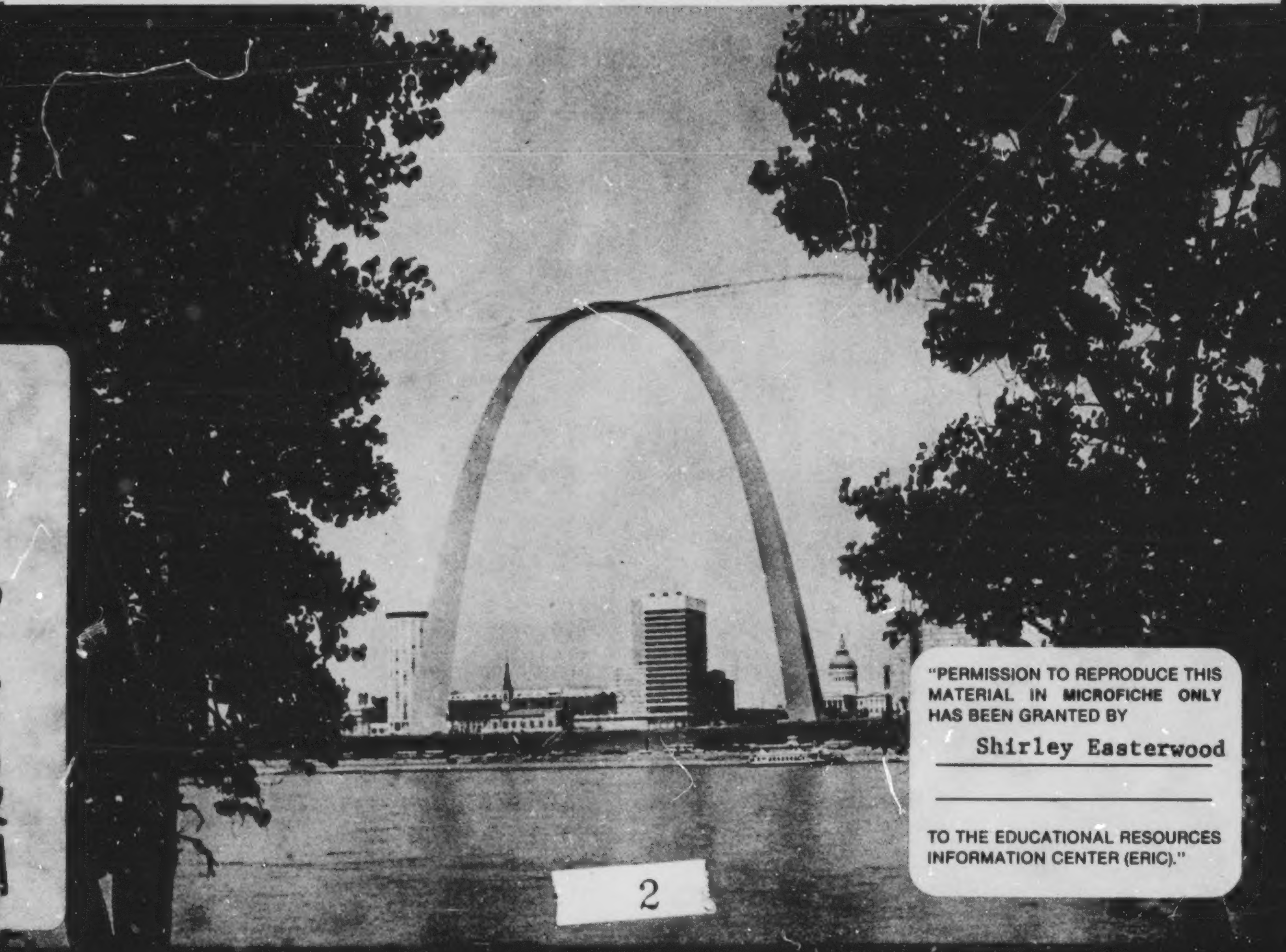
PROCEEDINGS

18th ANNUAL CONVENTION

"A Gateway To The Use of Computers In Education"

April 13 - 16
1980

Chase Park Plaza
St. Louis, Missouri



IR008630

"PERMISSION TO REPRODUCE THIS MATERIAL IN MICROFICHE ONLY HAS BEEN GRANTED BY
Shirley Easterwood

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

Additional copies available for
$12.00, prepaid, from:

Association for Educational Data Systems
1201 Sixteenth Street, N.W.
Washington, D. C. 20036

Library of Congress Catalog Card Number ISSN 0147-9296

Copyright (c) 1980, Association for Educational Data Systems

GATEWAY ARCH

Soaring high above the city of St. Louis, the stainless steel Gateway Arch presides over the waterfront and cuts a path of glory 630 feet tall into the sky. Designed by the late and noted architect, Eero Saarinin, the monument memorializes the spirit of the pioneers who settled the West after the Louisiana Purchase. Sixteen windows on each side of the Observation Deck at the apex afford breathtaking panoramic views to all directions. The transporter system to the top of the Arch operates from one hour after opening time to one half hour before closing time.

Photography courtesy: St. Louis Convention & Visitors Bureau
500 N. Broadway
St. Louis, Missouri 63102

PREFACE

The 1980 AEDS Convention is the Eighteenth Annual Convention of the Association. The Association for Educational Data Systems is a private non-profit educational organization founded in 1962 by a group of professional educators and technical specialists in educational applications. Its intention is to provide a forum for the exchange of ideas and information about the relationship of modern technology to modern education.

AEDS membership is open to all persons interested in learning more and keeping informed about current developments and directions in educational data systems and applications of computer technology to education. Included in the membership are educators and technical experts from all over the United States, Canada and other countries and representing public and private schools, higher education, state and provincial departments of education, the federal government, the armed forces and other professional and technical groups interested in educational computer applications.

The theme for this year's convention "A Gateway to the Use of Computers in Education" in many respects captures the theme of AEDS. The 1980 convention features a large program representing primary, secondary, and college level educators. The program is subdivided into the Technical Program and the Vendor Sessions.

The Technical Program includes a full slate of contributed, refereed papers submitted for publication in these Proceedings. The technical papers have been grouped as much as possible to reflect common broad areas of interest. There has also been an attempt made to avoid scheduling concurrent technical sessions involving similar areas of interest.

The Vendor Sessions offer a unique opportunity to gain first-hand information on products available from the commercial environment.

The convention team has endeavored to give you as much time as possible to visit with colleagues in areas related to your primary interest. Within the constraints of a three-day conference certain compromises were required but we sincerely hope you will have the time to visit and learn from colleagues as well as attend the technical and vendor sessions.

The process of creating a program of this magnitude involves the hard work and cooperation of a great many individuals. We are sincerely grateful to all the authors, chairpersons, session organizers, and others who contributed so generously of their time and effort to make the program possible.

It is you, the conference attendees, who in the final analysis will make the conference a success. To that end we wish to thank all of those individuals who attended the 1980 AEDS Convention and who have contributed to its success.

Arlan R. DeKock
1980 Program Chairman

Ralph E. Lee
1980 Convention Coordinator

TABLE OF CONTENTS

page

Preface iii

Table of Contents iv

1980 Convention Team vii

Convention Advisory Committee (1979-80) viii

AEDS Board of Directors (1979-80) ix

Committee Chairs (1979-80) x

Affiliate Group Presidents (1979-80) xi

Representatives To: (1979-80) xii

Sustaining Members of AEDS xiii

PAPERS

Bailey, T.E., Webster, R.E.	A Testing and Record Maintenance Package for Large Classes	1
Beck, James D.	Some Simple But Effective Ways to Use the Computer in Chemistry Instruction	5
Bell, Fred	CAI and Computer Literacy A Ten Year School/ University Project	8
Best, Alvin M., Pedersen, Neil G.	A Computerized System for Projecting Student Membership in School Systems Experiencing Dramatic Changes in Their Student Assignment Plan	15
Brebner, Ann, Hallworth, H.J.	A Multi-Media CAI Terminal Based Upon a Microprocessor With Applications For the Handicapped . . .	18
Brebner, Ann, Hallworth, H.J., McIntosh, Esther, Wontner, Cecil	Teaching Elementary Reading by CMI and CAI	21
Blood, Robert, Serena, Michael, Goldberg, Myron	A Computer-Based Management Program For Selection of Applicants to Allied Health Educational Programs	26
Boyson, Vicki A., Thomas, Rex A.	Interaction of Cognitive Style with Type of Feedback Used in Computer--Assisted Equation Solving	31
Bozeman, William C., Spuck, Dennis W.	A Design for the Evaluation of Management Information Systems	37

Brammer, Norman D.	Database Processing: Teaching Via An Instructional Data Base System	42
Brillhart, L.V., Barnett, A.	Humanizing the Introduction to Computers	47
Burson, Jeanne	"Standard" Review Units In a CAI Materials Appraisal Process	52
Contolini, Judith	An On-Line Instructor Controlled Management & Reporting System with Selectable "Tracks"	58
Cross, B.D., Barbay, Joseph E., Besterfield, Dale H.	Micros-By-The-Dozen	62
Denenberg, Stewart A.	Teaching Semantic Information Networks Using Microcomputers	65
Donoian, George	Humanistic Computer Programming: A Gateway to Learning	70
Engel, Gerald L.	Computer Science Education in Small Colleges: The Current Status	74
Horton, Richard E.	Using Personal Computers As Terminals In Computer Networks	80
Huntington, J.F.	Can The User of a Large Time Sharing System be Satisfied with a Microcomputer?	84
Jokela, Willis	Microcomputers in the Elementary School: What Experience has Taught Us So Far	87
Jones, Wayne A., Seemann-Jones, Marilyn	Management Information Systems - Programming For Individuals with Exceptional Educational Needs . .	89
Kehler, Thomas P., Barns, Mike	Design Alternatives for Conversational Consultation Systems	95
Kehoe, Daniel K.	New Administrative Software From Data Base Technology	101
Kosel, Marge	Designing Educational Microcomputer Software: A Model in Use	104
L'Allier, James J., Tennyson, Robert D.	A College Level Introductory Course On Computers in The Classroom: A Case Study	108
Lewis, Meg, Villiers, Alan, Rudnick, Betty	Use of the Quest Test Management System in a Milti-Campus Environment.	119
Mann, Don	Instruction of Applied Mangement Science Techniques in a Two-Year Business Programming Curriculum . . .	123
Marino, Phillip P.	Comprehensive School Facilities Inventory System .	125
Marshall, Sr. Patricia, Alderman, Donald L., Jaeger, Richard	Computing In Minority Institutions: An Update . .	130
McAdams, Sr. Joseph K.	Computer Literacy: A Workshop Approach for High School Math Teachers	135
McKinley, Robert L., Reckase, Mark D.	Computer Applications To Ability Testing	137
McHenry, Hugh	Using Computerized Data Management to Increase Student Responsibility for Learn in College Mathematics	142

Molettiere, Lou, Konsynski, Benn, Scott, Jack	Development of a Distributive Education System	149
Molloy, Larry	Distributed Data Processing at the Wayne County Intermediate School District	152
Mondy, Judy Bandy	Instructional Computing in Texas Public Schools-Applications and Administration	155
Orr, Norman, Zaccagnini, Gary	The Courseware Developmental Cycle as a Model for Planning and Implementing Automated Training Systems for an On-Line Claims Processing System	159
Pedersen, Neil G.	Employing Data-Management Packages To Modernize Your Planning Department	161
Roblyer, M.D.	More Hands for Teachers: Report of an Instructional Computing Study For the State of Florida	165
Ryan, Charles W., Drummond, Robert J. Bilodeau, Gerard	Differential Impacts of a Computer Information System on Selected Human Service Agencies	172
Schrage, John F., Sharp, John A.	The Computer Laboratory	175
Sheflin, Richard A.	The Development of a Multi-Cultural, Computer Assisted Reading Program	181
Slater, N. Michael	Five Steps for Acquiring Your Next Computer	187
St. John, Edward P.	Computer Policy in Higher Education: Major Statewide Issues	192
Sturdivant, Patricia	Microcomputers--Promoting Their Use in Elementary and Secondary Schools	200
Whitt, Jerry D., Adams, C. Nathan	The Usage of Short Internal Sort Algorithms for Teaching Mulitple Approaches to Elementary Computer Functions	207

1980 AEDS CONVENTION TEAM

CONVENTION COORDINATOR

Ralph E. Lee
Director of Computer Center
University of Missouri-Rolla
Rolla, MO 65401
(314) 341-4841

ASSISTANT CONVENTION COORDINATOR

John W. Hamblen
Chairman of Computer Science
University of Missouri-Rolla
Rolla, MO 65401
(314) 341-4491

PROGRAM CHAIRPERSON

Arlan R. DeKock
Associate Professor of Computer Science
University of Missouri-Rolla
Rolla, MO 65401
(314) 341-4496

PUBLICITY CHAIRPERSON

Sally Gaebler
Scientific Programmer/Analyst II
Computer Center
University of Missouri-Rolla
Rolla, MO 65401
(314) 341-4841

FACILITIES AND EXHIBITS CHAIRPERSON

Daniel C. St. Clair
Associate Professor of Computer Science
UMR Graduate Engineering Center
8001 Natural Bridge
St. Louis, MO
(314) 553-5432

FACILITIES AND EXHIBITS ASSISTANT CHAIRPERSON

Tony Iovino
McDonnell Douglas Automation Company
Suite 1100, 111 Westport Plaza
St. Louis, MO 63141
(314) 232-8091

AUDIO-VISUAL AND SPECIAL EQUIPMENT

John H. Eliff, Director of Data Systems
St. Louis Community College
5801 Wilson Avenue
St. Louis, MO 63110
(314) 644-9578

MEAL FUNCTIONS AND RECEPTIONS

Marvin Hahn
Hazelwood School District
15955 New Halls Ferry Road
Florissant, MO 63031
(314) 921-4450 ext. 256

OUTSIDE ACTIVITIES PROGRAM

Bill Heinbecker
Director of Computer Center
University of Missouri-St. Louis
St. Louis, MO 63121
(314) 553-5131

FINANCE CHAIRPERSON

David W. Dearth
Associate Director of Computer Center
University of Missouri-Rolla
Rolla, MO 65401
(314) 341-4841

MISSOURI AEDS LIAISON

Harry Thompson
1916 Elm
St. Charles, MO 63301

EVALUATION

Mr. Dale Schneiderhan
MECC-MIS
2021 E. Hennepin Avenue L140
Minneapolis, MN 55413

CONVENTION ADVISORY COMMITTEE

CHAIRMAN

Mr. Arthur Daniels **

1980 CONVENTION

Mr. Ralph Lee
Box 951
Rolla, MO 65401
314/341-4841

1981 CONVENTION

Mr. Dale Schneiderhan
MECC-MIS
2021 E. Hennepin Avenue LL40
Minneapolis, MN 55413

1982 CONVENTION

Mr. Randall Johns
Director, Data Processing
202 E. Commercial Street
Sanford, FL 32721
305/322-1252

WORKSHOPS - REGIONAL

Mr. Eugene Collins
Jefferson Co. Pub. School
809 Quail Street
Lakewood, CO 80215
303/988-5810

Mr. Thomas Hartsig
Computer Science
Macomb Intermediate SD
44001 Garfield
Mt. Clemens, MI 48044
313/286-8800

Mr. Ralph Lee
Computer Center
University of Mo.-Rolla
Rolla, MO 65401
314/341-4841

AEDS BOARD OF DIRECTORS

1979-80

PRESIDENT

Donald C. Holznagel
Coordinator, MicroSIFT
NWREL
710 S.W. Second Avenue
Portland, OR 97204
503/248-6974

PRESIDENT-ELECT

Dr. Winston Addis
Supt. of Schools
Mt. Vernon Comm. Schools
Mt. Vernon, IA 52314
319/895-8845

PAST PRESIDENT

Mr. Brad Burris
Editorial Vice President-
Testing
Scott, Foresman and Company
1900 E. Lake Avenue
Glenview, IL 60025
312/729-3000

SECRETARY

Dr. Bobby Brown (1981)
Computer Center
University of Iowa
Iowa City, IA 52242
319/353-3170

TREASURER

Mr. Steve Raucher (1982)
Montgomery Co. Public Schools
850 Hungerford Drive
Rockville, MD 20850
301/279-3581

****

Mr. Bruce Alcock (1981)
Riverdale Country School
S. Fieldston Road
W. 253 Street
Bronx, NY 10471
212/549-8810

Dr. Don Costello (1980)
Director, Academy Computer Srvs.
University of Nebraska
3835 Holdrege
Lincoln, NE 68583

Mr. Arthur Daniels, Jr. (1980)
RAMS Cooperative
31201 Dorchester
Madison Heights, MI 48071
313/585-7530

Mr. Charles Fitzgerald (ex officio)
Senior Systems Analyst
Data Processing Division
Saint Louis Public Schools
911 Locust Street
St. Louis, MO 63101
314/231-3720

Ms. Carolyn Meinhardt (1982)
ESV Region III
570 1st Street, SE
St. Cloud, MN 56301
612/255-9440

Mr. John Kelly (ex officio)
Palm Beach Junior College
4200 Congress Avenue
Lake Worth, FL 33460
305/965-8000

Dr. D. Thomas King (1980)
St. Paul Public Schools
360 Colborne Street
St. Paul, MN 55102
612/298-5867

Dr. Dale LaFrenz (1980)
2006 E. Lillian Lane
Arlington Hgts., IL 60004

Sister Patricia Marshall (1980)
Director, Computer Center
Xavier University
7325 Palmetto Street
New Orleans, LA 70125
504/486-7411 Ext. 255

AFIPS Representative (ex officio)
Dr. Judith Edwards
NW Regional Education Lab
710 SW 2nd 400 Lindsay Blvd.
Portland, OR 97204
503/248-6858

EXECUTIVE SECRETARY

Ms. Shirley Easterwood
1201 16th Street, N.W.
Washington, D.C. 20036
202/833-4100

COMMITTEE CHAIRS

1979-80

AWARDS

Mr. Jack Hoffman
Dir., SEPARS
Len B. Jordan Bldg., Rm 343
650 W. State Street
Boise, ID 83720
208/384-3236

BYLAWS AND CONSTITUTION

LONG RANGE PLANNING

Dr. Richard Munsterman
College of Education
Western Michigan University
Kalamazoo, MI 49008

MEMBERSHIP

Dr. Winston Addis **

NOMINATIONS

Mr. Dean Crocker
Grimes State Office Bldg.
Statewide Computer Service
Des Moines, IA 50819
515/281-4802

PROGRAMMING CONTEST

Mr. Tom Schmeltzer
Oakland Schools
2100 Pontiac Lake Road
Pontiac, MI 48054

BUSINESS AND INDUSTRY

HISTORIAN

Mr. Brad Burris **

POLICIES AND PROCEDURES

Dr. John Haugo
Executive Director
MECC
2520 Broadway Drive
St. Paul, MN 55113

PUBLICATIONS

Dr. Jerry Foecke
Executive Director
ESV Region III
570 First St., S.E.
St. Cloud, MN 56301
612/255-9440

MONITOR

Dr. D. Thomas King **

JOURNAL

Dr. Dennis Spuck
Associate Professor & Director
Education Research Center
AED/FH University of Houston
Houston, TX 77004
713/749-1667

BULLETIN

Ms. Carolyn Heding **

PUBLICITY

Mr. Conrad Goldberg
Psychological Corp.
757 Third Avenue
New York, NY 10022

**See Board of Directors for address

AFFILIATE GROUP PRESIDENTS

Functional:

COUNCIL OF REGIONAL AND
EDUCATIONAL ADMINISTRATIVE
MANAGERS (CREAM)

John Manoni (10/79-9/80)
Orange County Schools
Box 271
Orlando, FL 32801

Geographic:

ARIZONA

Lotus M. Knief (4/78-4/80)
College of ED. 525
University of Arizona
Tucson, AZ 85721

FLORIDA

Michael Staggs (6/79-5/80)
Orange City Schools
3225 Drake Drive
Orlando, FL 32810
305/423-9243

ILLINOIS

Roger Ault (11/79-10/80)
Coord. of Information Sys.
Township High School Dist.113
1040 Park Avenue West
Highland Park, IL 60035
312/432-6510

IOWA

Marion Hubers (11/79-10/80)
Iowa Lake Community College
Estherville, IA 51334
712/362-2604

MANITOBA

Jim Stuparyk (9/79-8/80)
Churchill High School
510 Hay Street
Winnipeg, Manitoba R3L 2L6
204/474-1301

MICHIGAN

Don Mann (10/79-9/80)
Oakland Comm. Col.
27055 Orchard Lake Road
Farmington, MI 48024
313/476-9400

MID-AMERICA

Kenneth Mendenhall
(6/79-6/80)
Hutchinson Comm. Col.
1300 N. Plum
Hutchinson, KS 67501
316/663-5781

MID-SOUTH

Walter Mathews
School of Ed., Suite 150
University, MS 38677
601/232-7195

MINNESOTA

Rosemary Schneiderhan
(10/79-9/80)
Roseville Pub. Schools
1251 W. County Rd B2
Roseville, MN 55113
612/633-8150

MISSOURI

Harry Thompson (5/79-5/80)
1916 Elm
St. Charles, MO 63301

NEBRASKA

Dirk Dierkhising (5/79-5/80)
Omaha Public Schools
4469 Farnam
Omaha, NE 48153
402/554-6192

NEW JERSEY

James Megletti
Teaneck Public Schools
1 West Forest Avenue
Teaneck, NJ 07666
201/837-2232

NEW YORK

Mary Heagney (10/79-9/80)
9201 Shore Road
Brooklyn, NY 11209
212/596-5850

NORTH CAROLINA

John Beegle (9/79-8/80)
Western Carolina Univ.
Accounting & Infor. Systems
Cullohowhee, NC 28723

OHIO

Robert McIlvain
Toledo Public Schools
Manhattan & Elm
Toledo, OH 43608
419/729-5111

OREGON

Charles G. Hadduck (1/80-12/80)
501 N. Dixon
Portland, OR 97228
503/249-3000 x512

TEXAS

Donald Offerman (10/79-9/80)
Texas Education Agency
201 E. 11th Street
Austin, TX 78701
512/475-2259

VIRGINIA

Charles Richardson (6/79-6/80)
MASD
8th St. Office Bldg.
Richmond, VA 23233
804/786-7725

WISCONSIN

John C. Bernard (5/80-4/81)
Kenosha Unified SD #1
625 - 52nd Street
Kenosha, WI 53141
414/656-6347

REPRESENTATIVES TO:

1979-80

AFIPS

Dr. Judith Edwards
NW Regional Education Lab
710 SW 2nd 400 Lindsay Blvd.
Portland, OR 97204
503/248-6858

AFIPS EDUCATION COMMITTEE

Mr. Don Holznagel **

AASA

Dr. Winston Addis **

AERA

Dr. Leroy J. Tuscher
Graduate School of Education
Division of Ed Adm
Lehigh University
Bethlehem, PA 18015

ASBO

MINORITY INSTITUTIONS

Sr. Patricia Marshall **

JECC

Mr. Don Holznagel **

ICCP

Mr. Paul Pair
Control Data Corp.
1650 W. Alameda Dr., Suite 150
Tempe, AZ 85282
602/968-8791

Mr. Roland Spaniol
Dir., Data Processing
Eastern Illinois University
Charleston, IL 61920
217/581-3227

NCTM

Mr. J. Richard Dennis
University of Illinois
Education Building
Urbana, IL 61801
217/333-4664

NCME

Mr. Brad Burris **

SALT

Mr. Steve Raucher **

CUCC/NECC 2

Dr. Bobby Brown **

DECUS

Mr. Bruce Alcock **

** See Board of Directors for address

SUSTAINING MEMBERS OF AEDS

Arthur Andersen and Company, Chicago, IL

Automated Data Entry Systems, Williamstown, NJ

Computer Curriculum Corporation, Palo Alto, CA

Management Science America, Atlanta, GA

MICRO Power and Light, Dallas, TX

National Computer Systems, Minneapolis, MN

Tektronix, Inc., Beaverton, OR

A TESTING AND RECORD MAINTENANCE PACKAGE FOR LARGE CLASSES

T. E. Bailey
Oklahoma State University

R. E. Webster
Standard Oil of Indiana
Tulsa, Oklahoma

Introduction

The Department of Computing and Information Sciences of The Oklahoma State University offers an introductory programming course which has an enrollment of about 800 students in a number of theory and laboratory sections. Regular faculty and teaching assistants comprise the instructional staff. In an effort to maintain as much continuity and consistency as possible among sections and instructors, there are ten theory sections meeting once a week and containing up to 100 students each in which videotaped lessons are presented. The students are placed into 22 smaller laboratory sections of about 40 students each twice a week so that they can interact directly with a "live" instructor. In the lab sections assignments are passed out and taken up, and all testing is done. Because of this, all records for purposes of grading are kept by lab sections. Unfortunately, the Registrar requires that grades be reported for official university records by theory sections. Just keeping track of enrollments in such a large course is difficult; but when records in the course are organized in one way and the final grades must be reported based upon an entirely different organization, the bookkeeping problem becomes horrendous.

For many years the Department had been using computer-generated and graded testing in the course, and subsequently a computer program for reporting final grades by lab sections had been operational, along with a program for punching JOB cards and a program for assigning rooms and seats for tests. A master file on punched cards, each of which contained enrollment information for one student, was used as input for each of the programs; and the master file was maintained by hand with the help of a card sorting machine. Of course, a much nicer way would be to use an on-line disk file that could be accessed in any one of several possible orders and that could be used both as input to the various administrative programs and for output from the test grading program. Such a file would make it possible for records to be kept by lab sections and theory sections simultaneously, satisfying both the needs of the instructional staff and the Registrar.

During the Spring Semester of 1978 the specifications for the file and attendant programs were drawn up. Each record must contain, for one student, enrollment and identification information, test and programming assignment results, final grade information, and a remarks field. The file must be accessible in alphabetical order, in alphabetical order by theory sections, in alphabetical order by lab sections, and in order by randomly assigned four-digit numbers (so that records could be posted without names and student ID numbers to protect privacy). Furthermore, the file must be compatible with the existing programs, if possible. Additional programs must be produced for generating, updating, and listing the file.

The programs were implemented in PL/I on an IBM System/370, Model 158, with MVS operating system. The package became operational in the Fall Semester of 1978, and is now operating on a System /370, Model 168, with MVS.

Description of the Student File and the Maintenance Program

The file is a triply-linked list containing a header record and one record for each student. The header record contains the course identification (name, number, date, term--any 18 characters); eight fields for possible points, corresponding to the eight fields for test and assignment results in a student record; pointers to the first record for each of three orders, alphabetical by primary (theory, for example) sections, alphabetical by secondary (lab, for example) sections, and increasing order on the four-digit randomly assigned code number; primary section name; and secondary section name. A student record contains three pointers, name, student ID number, college, classification, sex, four-digit code number, primary section number, secondary section number, eight fields for recording test and assignment results and a flag associated with each field, total points accumulated, a flag for dropping the student, and a remarks field. The name, student ID number, college, classification, and sex are obtained from enrollment data provided on punched cards from the Registrar. The pointers are assigned by the maintenance program during generation of the file, as is the four-digit code number, which is a pseudo-random number produced by a very simple algorithm. The eight results fields are initialized to negative one and their associated flags to "off," the total points field to zero, the remarks field to blank, and the drop flag to "off."

The maintenance program has a number of capabilities besides generating the file. A record may be added to or deleted from the file. A student may be dropped from the file by setting the drop flag to "on." (A student may be dropped from the file in two ways, by deletion and by the drop flag. When a student record is deleted, of course it can no longer be accessed. On the other hand, when the drop flag is "on," that record is accessible and may be listed with a "dropped" message.) The maintenance program can also change any field in the student record except the student ID number, the code, the remarks field, and any grading information. In addition, one of several types of sorting routines may be specified for linking records upon generation of the file.

Description of the Update and Report Programs

The update program has the capability of changing any field which contains test and assignment results, the remarks field, and the fields containing the possible points in the header record. It can also change a flag associated with a results field to "on," indicating that the student has been excused from that particular test or assignment. (It can also turn an excused flag "off.")

The results of all hand graded materials in the course are entered into the file by the update program using punched cards containing the pertinent information. These cards are produced by a separate program that punches all of the required information for updating the file, one card for each student, except for the results that are to be entered, of course. Instructors must punch the results.

The report program prints any of several listings of the file and appends to each record printed a percentage score, calculated from the points earned and the possible points at the time of listing, and a letter grade based upon a grading scale provided as input to the program. Four basic listing options are available. Three are provided by the triply-linked list organization: (1) alphabetical by primary (theory) sections used primarily for reporting grades to the Registrar; (2) alphabetical by secondary (lab) sections, used primarily by individual lab instructors; and (3) in order by four-digit code number, used for posting current and final grades so that students can examine their grades. (Names and student ID numbers are not printed on this report; each student knows his own four-digit code, but cannot identify the grade of any other particular student, fulfilling the legal requirement of protecting the students' privacy.) A fourth listing in alphabetical order is obtained by performing a sort on a copy of the file and then listing it, used for ready reference by the supervising instructors and for the permanent records kept for the course. Each report contains for each record printed the name and student ID number (both omitted when records are ordered on the four-digit code), the four-digit code, primary and secondary section numbers, each of the fields for test and assignment results for which there is a corresponding value greater than zero in the header record, the accumulated points, the percentage of total points accumulated, a letter grade, and the contents of the remarks field. (See Figures 1 and 2 for sample pages from the report program.) If a results field flag is "on," indicating an excused assignment or test, EXC is printed in the respective field and the percentage is computed using the possible points less the point value of the excused test or assignment. If a field contains a negative one in the record, indicating that the field is "empty" because the student did not take the test or turn in the assignment, a hyphen is printed in the respective field. If the drop flag of a record is "on," then a dashed line is printed through all of the numeric fields used and the letter grade field; the contents of the remarks field are still printed.

It is possible to get reports ordered on any one, two, or three fields of the records. For example, an alphabetical listing of all female students followed by an alphabetical listing of all male students could be obtained. Or alphabetical listings by colleges in alphabetical order could be obtained. One other special listing can be obtained for mid-term grade reports, listing only freshmen in alphabetical order by primary or secondary sections.

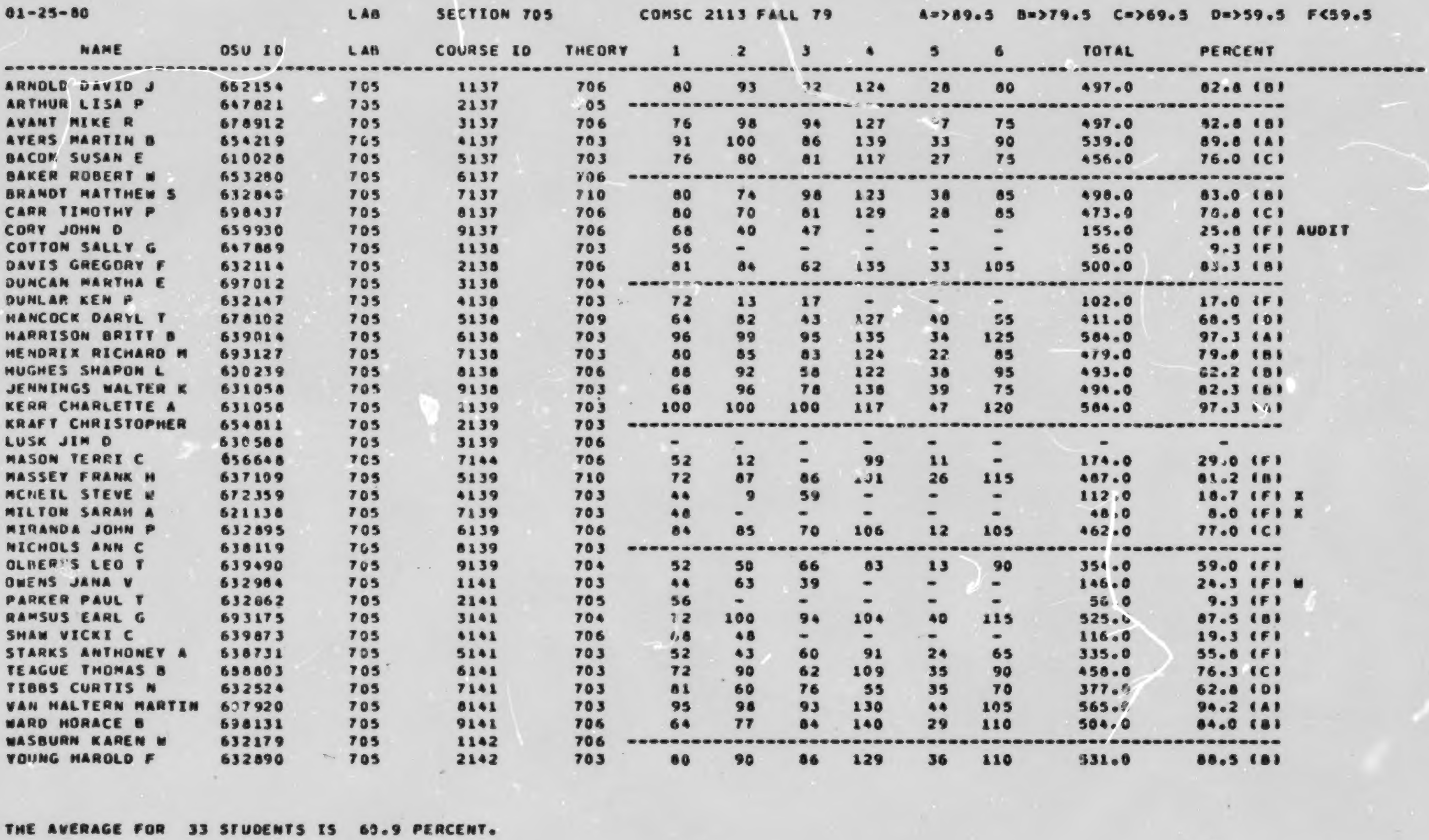
Test Generation and Grading Programs

The test generating program produces a multiple-choice test for each student in the file for whom the drop flag is "off." The tests are printed alphabetically by primary or secondary sections; each test contains a name, the corresponding four-digit code, a test number, any instructions, and a set of questions.

Tests may be generated so that all tests are alike or all different. For the latter case, there are three options: (1) the same set of questions may simply be printed in random order, (2) alternating tests may contain all even-numbered or all odd-numbered questions, or (3) some randomly selected subset of questions may be printed. The third option allows considerable flexibility. For example, suppose that there are 40 questions, but each test is to have 25 questions. Then 25 questions may simply be selected randomly from the 40 questions. But it is also possible to select random subsets of subsets. Thus, a subset of 25 questions might also be obtained by randomly selecting 6 from the first 12, 15 from the next 20, and 4 from the last 8.

For each test generated a record is written onto a "test key" file in which the number, in the original set of questions, of each question chosen is placed in order as it appears on the test. For example, for 10 questions in random order, a test key might be 8 6 10 3 4 9 7 2 5 1. The test keys are, of course, necessary for grading the tests.

Each student's responses are punched, along with his name, test number, and four-digit code. (These cards are punched by another program, except for the responses, which are punched by hand.) Alternately, OMR cards could be used. These cards, in any order, are used as input to the test grading program which, using the test key file, grades each test and produces a listing ordered on the four-digit code numbers of test results and the correct responses for each test, which is posted for student use. An alphabetical listing of results by primary or secondary sections is also produced for use of the instructors. Finally, an overall summary of results (that is, the distribution of responses for each question) and summaries for individual sections are printed. The test grading program also updates the student file, placing each student's score into the appropriate



```
01-25-80                    LAB    SECTION 705      COMSC 2113 FALL 79       A=>89.5  B=>79.5  C=>69.5  D=>59.5  F<59.5

      NAME            OSU ID      LAB     COURSE ID   THEORY    1     2     3     4     5     6        TOTAL        PERCENT
------------------------------------------------------------------------------------------------------------------------------
ARNOLD DAVID J        662154      705       1137       706     80    93    92   124    28    80        497.0       82.8 (B)
ARTHUR LISA P         647821      735       2137       705   ------------------------------------------------------------
AVANT MIKE R          678912      705       3137       706     76    98    94   127    27    75        497.0       82.8 (B)
AYERS MARTIN B        654219      765       4137       703     91   100    86   139    33    90        539.0       89.8 (A)
BACON SUSAN E         610028      705       5137       703     76    80    81   117    27    75        456.0       76.0 (C)
BAKER ROBERT W        653280      705       6137       706   ------------------------------------------------------------
BRANDT MATTHEW S      632840      705       7137       710     80    74    98   123    38    85        498.0       83.0 (B)
CARR TIMOTHY P        698437      705       8137       706     80    70    81   129    28    85        473.0       78.8 (C)
CORY JOHN D           659930      705       9137       706     68    40    47     -     -     -        155.0       25.8 (F) AUDIT
COTTON SALLY G        647889      705       1138       703     56     -     -     -     -     -         56.0        9.3 (F)
DAVIS GREGORY F       632114      705       2138       706     81    84    62   135    33   105        500.0       83.3 (B)
DUNCAN MARTHA E       697012      705       3138       704   ------------------------------------------------------------
DUNLAR KEN P          632147      735       4138       703     72    13    17     -     -     -        102.0       17.0 (F)
HANCOCK DARYL T       678102      705       5138       709     64    82    43   127    40    55        411.0       68.5 (D)
HARRISON BRITT B      639014      705       6138       703     96    99    95   135    34   125        584.0       97.3 (A)
HENDRIX RICHARD M     693127      705       7138       703     80    85    83   124    22    85        479.0       79.8 (B)
HUGHES SHAPON L       630239      705       8138       706     88    92    58   122    38    95        493.0       82.2 (B)
JENNINGS WALTER K     631058      705       9138       703     68    96    78   138    39    75        494.0       82.3 (B)
KERR CHARLETTE A      631058      705       1139       703    100   100   100   117    47   120        584.0       97.3 (A)
KRAFT CHRISTOPHER     654811      705       2139       703   ------------------------------------------------------------
LUSK JIM D            630588      705       3139       706      -     -     -     -     -     -           -            -
MASON TERRI C         656648      705       7144       706     52    12     -    99    11     -        174.0       29.0 (F)
MASSEY FRANK H        637109      705       5139       710     72    87    86   101    26   115        487.0       81.2 (B)
MCNEIL STEVE W        672359      705       4139       703     44     9    59     -     -     -        112.0       18.7 (F) X
MILTON SARAH A        621138      705       7139       703     48     -     -     -     -     -         48.0        8.0 (F) X
MIRANDA JOHN P        632895      705       6139       706     84    85    70   106    12   105        462.0       77.0 (C)
NICHOLS ANN C         638119      765       8139       703   ------------------------------------------------------------
OLBERGS LEO T         639490      705       9139       704     52    50    66    83    13    90        354.0       59.0 (F)
OWENS JANA V          632984      705       1141       703     44    63    39     -     -     -        146.0       24.3 (F) W
PARKER PAUL T         632862      705       2141       705     56     -     -     -     -     -         56.0        9.3 (F)
RAMSUS EARL G         693175      705       3141       704     72   100    94   104    40   115        525.0       87.5 (B)
SHAW VICKI C          639873      705       4141       706     68    48     -     -     -     -        116.0       19.3 (F)
STARKS ANTHONEY A     638731      705       5141       703     52    43    60    91    24    65        335.0       55.8 (F)
TEAGUE THOMAS B       698803      705       6141       703     72    90    62   109    35    90        458.0       76.3 (C)
TIBBS CURTIS N        632524      705       7141       703     81    60    76    55    35    70        377.0       62.8 (D)
VAN HALTERN MARTIN    607920      705       8141       703     95    98    93   130    44   105        565.0       94.2 (A)
WARD HORACE B         698131      705       9141       706     64    77    84   140    29   110        504.0       84.0 (B)
WASBURN KAREN W       632179      705       1142       706   ------------------------------------------------------------
YOUNG HAROLD F        632890      705       2142       703     80    90    86   129    36   110        531.0       88.5 (B)


THE AVERAGE FOR  33 STUDENTS IS  63.9 PERCENT.
```

Figure 1. Sample page of report ordered alphabetically by lab sections. Only six of the eight fields for test results or assignments were used.

```
COURSE ID      1      2      3      4      5      6        TOTAL       PERCENT
---------------------------------------------------------------------------------------------
   6184     ------------------------------------------------------------------------
   6185     ------------------------------------------------------------------------
   6186       73     80     80     60     33    110        436.0       72.7 (C)
   6187     ------------------------------------------------------------------------
   6188       56     69     60    138     15     75        413.0       68.8 (D)
   6189       96     96     95    137     37    100        561.0       93.5 (A)
   6191       84    100     98    131     37    125        575.0       95.8 (A)
   6192       64     64     72    106     23     70        399.0       66.5 (D)
   6193       66     80     58    112     21     90        427.0       71.2 (C)
   6194       98     86     94    123     37    125        563.0       93.8 (A)
   6195     ------------------------------------------------------------------------
   6196       88     96     91    115     32    135        557.0       92.8 (A)
   6197       92    100     97    137     32    120        578.0       96.3 (A)
   6198     ------------------------------------------------------------------------
   6199       60     42     30      0      0     55        187.0       31.2 (F)
   6211       88    100     95    125     43    115        566.0       94.3 (A)
   6212       92     98     94    139     33    110        566.0       94.3 (A)
   6213       56     64     84    124     43     60        431.0       71.8 (C)
   6214      100    100     94    135     40    105        574.0       95.7 (A)
   6215       36     26     72     82     28     60        304.0       50.7 (F)
   6216       76     94     98    134     48    120        570.0       95.0 (A)
   6217      100    100    100    126     17    140        583.0       97.2 (A)
   6218       88     93    100    138     47    110        576.0       96.0 (A)
   6219       80     74     72    125     30     75        456.0       76.0 (C)
   6221       56     64     60     98     15      -        293.0       48.8 (F)
   7111        -      -      -      -      -      -          -           -
```

Figure 2. Sample report ordered on the four-digit code number.

field of the record, and puts the point value of the test into the header record.

Additional Programs That Make Use of the Student File

Other programs that use the student file as input are available for performing special functions. One makes room and seat assignments for students for taking tests. System JOB cards are punched by another. One performs a summary of grades and drops by college.

A new program which is being used for the first time this year was produced by S. R. Blevins (1). One of its functions is to provide additional protection of the computer account from unauthorized users. A student must have an active record in the student file before he is allowed final access to the computer. A second function is to encourage students to use their computer runs wisely by limiting the number of runs per week and also to prevent an excessive number of runs by individual students. A summary of student activity on the computer account is also available.

Conclusions

The package has met all of the design objectives and has operated as expected since the Fall Semester of 1978. The dreadful problem of final grade reports (as well as mid-term grade reports), which used to consume many laborious hours, has been eliminated. At the end of the Fall Semester of 1979, final grades were recorded on the Registrar's forms and on their way to the Registrar shortly after the final examination was given. The incidence of errors in recording final grades has also been practically eliminated.

Careful consideration was given in the design criteria to make the package general so that it would be applicable to any course with a large enrollment, and it has already been successfully implemented in one other such class.

References

1. Blevins, Stephen R. "Control of Student Computer Usage," M. S. Report, Oklahoma State University, 1979.

SOME SIMPLE BUT EFFECTIVE WAYS TO USE THE COMPUTER IN CHEMISTRY INSTRUCTION

James D. Beck
Department of Chemistry
Virginia State University
Petersburg, Virginia 23803

Although chemistry has been one of the main areas in which computer-based materials have been utilized for instructional purposes, the extent of use has really been quite limited. This despite the fact that the decreasing cost of computer hardware has made computer access a reality at nearly every institution of higher education and at many secondary schools. Why is it that most students are never exposed to computer-based learning, in chemistry or in other courses? I believe it is still due in part to the larger-than-life reputation that computers carry with them. These mighty monsters that are capable of such tremendous tasks have been viewed as complex black boxes which cannot be understood or used by normal folk. Instead they need to have highly-trained masters who can communicate with them in special languages and who are wizards at complex mathematics and electronics.

The shroud of mystique surrounding the computer has left faculty members (and students) awed and even fearful of the beast. It is difficult to get a college instructor to sit down at a computer terminal and try it out; it is nearly immpossible to convince ordinary teachers that they can and should develop computer-based instructional materials and use them in their classes.

The image problem has not been solved by demonstrations of CAI-oriented systems such as PLATO or by showing fancy graphics in multiple colors. While such spectacular demonstrations usually elicit immediate enthusiasm and even wonder, the effect is usually short-lived. When the potential user gets down to the reality of where, when, how, and how much, disillusionment sets in rapidly. He or she finds out that the cost of some CAI systems is exorbitant, that little software is commercially available, that what is available doesn't fit the school's program or students or won't run on the available hardware, and that anything that is to be done will have to be done by individual instructors with minimal assistance from "experts." The normal reaction is to say "forget it," at least for now. So for most college teachers and their students, computer-assisted instruction remains a "future maybe."

There are plenty of other valid reasons why the computer has not really had a significant impact on the instructional process as yet, but I think that the ones that I have just outlined play an important part in many of the negative decisions regarding computer utilization. Are they valid reasons in 1980? I think not, at least not for persons in the field of chemistry. There are meaningful ways in which the computer can be used to enhance learning. We need to convince high school and college teachers that they can overcome the obvious impediments and involve the computer in their instruction.

One way to start is by demonstrating some very simple types of computer use. If we can convince someone that even these very simple applications are effective, then computer use can start immediately and we can build from there.

At Virginia State University we have been using the computer in chemistry courses in a variety of ways over the past several years. Virginia State is a predominantly-Black, four-year, state-supported institution located in Petersburg, Virginia. Over 95 percent of the 3500 undergraduates are Black. We have never had any elaborate computing facilities; the computer has always been devoted largely to administrative functions. Several years ago we reached the level of interactive computing when seven IBM 2741 terminals were installed as part of a time-sharing system. These slow and unreliable relics carried us through until the past September when they were replaced by 20 crt units. Obviously, we now have greatly expanded capabilities. But even before we had interactive capability, we found some ways to utilize the computer in our classes.

We started out with programs for the analysis of laboratory data. There was nothing fancy about the programs and they didn't really do anything that couldn't be done without a computer, but they served to stimulate student and faculty interest. We ran these in batch mode initially, then converted the FORTRAN programs to BASIC for interactive use when our time-sharing system became operable.

With the interactive capability, we began to write some drill-and-practice programs in BASIC. Over the past three years we have gradually added programs to our library until we now have over 60 of them available for use by our students. Again, they are not sophisticated, having been written largely be a person who has had no formal training in programming. Despite this, they have been very well received by our students, including the large group of "underprepared" students. These tutorial and drill programs have definitely enhanced our instructional program by adding a different type of learning activity.

20

We have also tried our hand at simulations. These have been somewhat more difficult to develop than the tutorial programs, but we have worked up a few, often building on the ideas of other persons. These programs are also in BASIC and are run in the interactive mode.

Another area of use has been for the generation of individualized quizzes, examinations, and problem sets. This is nothing new; the computer has been used to generate repeatable examinations for many years. Most of the programs that are available for this purpose have been fairly complex and relatively difficult for the average instructor with no computer experience to use. We have written simple programs in BASIC and FORTRAN that allow us to generate the individual versions very easily. These programs do not have all of the flexibility and capability of their bigger relatives, but they get the job done for us with minimal effort and no special expertise.

While we have used the generating programs to turn out some individualized quizzes and examinations as part of an individualized study project, our main use has been for generating individual problem sets for off-line work by our students. These problem sets give us a great deal of flexibility and supplement our interactive programs. All types of questions can be included on the problem sets, whereas there are some significant limitations on interactive programs. Since the work in done off-line, we do not have to be concerned about problems that require extensive calculations, or about thought problems that would require an inordinate amount of connect time if done on-line. Even essay-type questions are not excluded. The sets are individualized by using random numbers and can be printed in batch mode or by each student separately. The answers can be generated along with the questions, if desired by the instructor.

We have been looking at these problem sets to see if they could be used for diagnostic-prescriptive purposes and have made some preliminary trials along this line. For this we have used a BASIC program to generate the problem sets and have printed the sets in batch mode for distribution to the students in class. The answers are generated at the same time that the questions are and are stored in a file. Each student then works out the answers to his or her set of questions, takes these to a terminal and accesses an interactive program that checks the answers. This program not only checks answers, but provides correct answers for any that the student gets wrong and provides detailed descriptions of the steps involved in solving these. In addition, the questions that were not done correctly are examined and specific areas of difficulty are identified. Additional activities are prescribed to remedy specific student problems. These include textbook readings, additional problems to solve, media programs, interactive computer programs, and conferences with the instructor or student tutors.

The diagnostic-prescriptive system is still in the formative stages, but we think that it shows enough promise to warrant investing more time and effort to implement it. So far it has been done only on a trial basis. There are a number of ways in which this type of system can operate, ranging from providing no feedback to the student to being a full-fledged system of computer-managed instruction. Following our past pattern, we are taking it slowly, experimenting with various things and doing it as simply as possible, recognizing the severe constraints of time, manpower, and money that we must continue to operate under.

I have been describing some of the ways in which we have employed the computer in chemistry instruction at a small minority institution that is not blessed with an abundance of money. My purpose has been to try to show that there are valid instructional uses which can be made of the computer despite the limitations of inadequate hardware, the restrictions of a general programming language such as BASIC, a lack of programming expertise, the inavailability of commercial software, and the limited time and money that most of us have available to develop and implement computer-based instructional materials. I have done nearly all of the things that I have described here on my own, with minimal assistance from computer personnel or other faculty members, no outside funding, no released time, and only an ocassional student worker to help with typing. Now that an increasing amount of software is becoming available, one should find it easier to get started in using the computer. New languages may be more useful for CAI purposes than a general one like BASIC. However, many worthwhile things can still be done with the ubiquitous BASIC. Similarly, new technology such as that permitting real graphics can add a tremendous dimension to CAI, but one need not wait for this to happen at your institution before getting involved with the computer. Tomorrow will undoubtedly bring more and better hardware and software and will make computer utilization much easier for all of us, but let's not wait until tomorrow when there are simple, but effective applications that can be made right now.

Having made the argument that everyone can do something with the computer, and having used our own situation as an example, the question of effectiveness still needs to be addressed. Even if there are instructional uses that can be made and one has been persuaded to try to implement some computer-based activities, is it worth the effort? I will not cite any extensive comparison studies or refer to the results of other users, but I will summarize our experiences at Virginia State. Our computer use has always been part of a broad instructional design which offers alternative ways to learn chemistry. This reflects my personal philosophy that the computer is only one of many instructional tools that are available to teachers and can be made available to students. We have used our computer-based materials in conjunction with other learning materials such as slide/tape programs, audio tapes, performance objectives, and study guides. And we have not omitted the traditional course components; students can still rely on a textbook, lectures, and recitation sessions to help them meet their personal needs.

One way in which we have evaluated our materials is to observe the extent to which they are used under our multiple alternatives system. The frequency of use data show that the interactive computer programs have been used very heavily. So have the computer-generated problem sets. Both of these types of computer-based materials have been used much more than other materials and activities that have been available to the students (slide/tape programs, audio tapes, practice sessions, etc.).

We have also conducted surveys to determine the students' perceptions of the effectiveness of the various alternatives in helping them to learn chemistry. The surveys show that the interactive computer programs and the computer generated problem sets are perceived as being very useful. However, the highest rankings of usefulness have invariably been given to the lectures and lecture notes. The computer-based materials have been rated significantly above the textbook, study guide, recitation sessions, suggested plans of study, media programs, and sets of objectives. The practice sessions and copies of old examinations are considered to be nearly as useful as the computer-related materials. Students have been nearly unanimous in desiring to have the system of alternatives and the computer-based materials continued.

There is a side benefit of computer use which cannot be directly measured. The computer can be an excellent motivating force, for both students and instructors. Most of the students who enroll in our first course in chemistry (General Chemistry) have never been directly involved with a computer. They are nearly always greatly impressed by it, sometimes to the point of fascination, and even when the use that is made of it is very simple. It appears that they spend more time and expend more effort on the computer activities than on more traditional learning activities, in part because of the novel nature of the computer. Many good arguments for computer use have been advanced--the interactive mode, the individualization, the immediate feedback, the infinite patience. Whatever the reasons, my students tend to like to use the computer programs and they think that they are using their time wisely and are learning something. The motivating effect may even carry over to other aspects of the course.

Instructors can also be motivated by getting involved with the computer. In my own case I credit using the computer with rejuvenating my teaching. I have enjoyed developing the programs and other materials and have frankly been quite thrilled by the positive student reaction to them. Despite the rather primitive nature of our computer-based learning activities, the results have been personally very satisfying. I recommend it to every college instructor. There is something that you can do with the computer; the rewards for both you and your students can be very real ones.

CAI AND COMPUTER LITERACY
A TEN YEAR SCHOOL/UNIVERSITY PROJECT

Fred Bell
University of Pittsburgh

Beginning in 1970, the University of Pittsburgh and Upper St. Clair School District developed and evaluated computer-literacy and computer-enhanced-learning programs. Emphasis was placed upon hands-on computer use ranging from drill and practice through problem solving for school and college students. A computer literacy course and a computer-enhanced learning curriculum were implemented in grades 7 through 12. Graduate and undergraduate courses on computer literacy and computer applications in society are taught to education majors at the University of Pittsburgh. Evaluations include profiles of students' abilities and attitudes in computer-enhanced learning environments, together with descriptions of hardware, course objectives, courseware, and student activities.

Just one generation ago the first electronic computer, the ENIAC, was put into operation at the University of Pennsylvania. It was a huge 100-foot long machine containing thousands of vacuum tubes, resistors, capacitors, and switches connected by miles of wires. It could add 10-digit numbers in 1/5000 of a second. Today, electronic computers the size of a small television set can do everything the ENIAC could do, and more, in a small fraction of the time it took the ENIAC.

By 1966, just 20 years after ENIAC, computer assisted instruction and computer literacy courses were being implemented in schools, colleges, and universities. For example, in 1967, the School District of Philadelphia introduced a computer literacy course for junior high students in five schools with the modern objectives:

1. To have the students understand the influence computers have on their lives.
2. To demonstrate how computers work and what they can and cannot do.
3. To begin to learn how to program the computer so that they can use the computer in their courses in the senior high schools.
4. To introduce students to the various careers which are available to them in the data processing field (Charp and Altschuler, 1976, p. 10).

By 1976 the computer literacy course became an integral part of the course offerings in 32 Philadelphia schools and computer assisted instruction has been used in Philadelphia in mathematics and reading courses since 1966. However, Philadelphia is a notable exception in the early uses of computers in schools. According to Andrew Molnar (1976), only 55 percent of secondary schools in the United States had access to computers in 1975 and only 23 percent of the schools were using their computers solely for instructional purposes.

Between 1955 and 1975 the United States was the major user and exporter of high technology (computer technology included) in the world; but now the United States is being overtaken in the development of new technologies by countries such as West Germany and Japan (Molnar, 1977). According to Molnar (1977):

> Today, USOE, NIE and NSF no longer have major programs designed to support computer innovations in education . . . With the exception of a modest basic research program in the Division of Mathematical and Computer Sciences, NSF has no major programs for computer innovation. While many federal programs for education do provide indirect support for computers through grants and contracts, there are no major programs designed to stimulate computer innovations or the uses of computers

in education (p. 41).

While public funding for computer literacy and computer uses in education has improved somewhat since 1977, there is little prospect that Federal agencies will, in the near future, provide the generous support needed to fund new projects such as PLATO, Project LOGO, TICCIT, Huntington II, and Project Solo. In spite of reduced Federal support for educational applications of computers, we need not fear that computers will disappear from schools. Dramatic new developments in microtechnology, with equally dramatic decreases in the cost of computer technology and the increased availability of good, inexpensive programs, are already stimulating the use of computers in businesses and homes. When most school students were able to afford a hand-held calculator, calculators became an integral part of many school mathematics programs. Now that microcomputers are inexpensive enough so that many families can purchase them for use in the home, this new technology is beginning to have a significant impact upon learning both at home and in school. In fact, computers and significant computer applications for most students in moderate-sized and large schools have been quite affordable for local school systems since 1970.

According to James Saunders (1978), director of the educational computing facility at Upper St. Clair High School, near Pittsburgh, Pennsylvania, the majority of graduating students (over 90 percent, in 1979) in this school system have been making significant uses of computers in the mathematics classes for the past ten years at a total average add-on cost of approximately $15,000 per year.

COMPUTERS AND EDUCATION IN THE SCHOOL AND UNIVERSITY

As part of a joint project, James Saunders, John DeBlassio, Chairman of the Mathematics Department at Upper St. Clair High School, and the writer of this paper who is Coordinator of Mathematics Education Programs at the University of Pittsburgh, have developed and implemented a joint school/university computer-enhanced learning, computer literacy, and computer-enhanced-learning research program. All of their work has been carried out through local funding by the school district and the university. The success of this project in integrating computer literacy, school mathematics, and problem solving was reported by Bell in the Summer 1979 AEDS Journal. This present paper is a report of the developmental work done jointly by the Upper St. Clair School District and the University of Pittsburgh from 1970 to 1980.

Computer Hardware Used in the Project

Following a brief period of renting computer time on a commercial time-sharing system, in 1971 Upper St. Clair High School purchased a DEC EDU20, PDP-8/E system with 8K of core memory, five teletype terminals, a card reader, and 32K of disk storage. In 1974 this system was upgraded to a DEC EDU25, PDP-8/E system with 24K of core and a DEC tape was added. In 1976 the system was again modified to an EDUCOMP ETOS, PDP-8/E with 32K of core memory, 8 terminals and 1600K of disk storage. The DEC PDP-8/E system now in use has 32K of core memory, an RK05 disk drive, a TC08 DEC tape control, five teletype terminals, two LA36 decwriters, one Tektronix 4006-1 video graphics terminal, one Intecolor 8001G color video graphics terminal and a CM8E mark-sense card reader. The system is controlled and operated by faculty members and is totally dedicated to educational uses by students and faculty in various subject areas. The system supports BASIC, FORTRAN, and PAL 8. Students have access to the computer lab before and after school, during free periods, and during certain classes.

Since 1973, the system used for the University of Pittsburgh part of this project has two DEC-1099 computers, about 100 public-access decwriter II terminals at several RJE stations, and a TRS-80 Level II computer with 16K of core memory and audio tape storage. RJE stations are open for student use from 8:00 a.m. to midnight seven days a week while the university is in session.

Modes of Computer Use in Teaching and Learning

Licklider (1979) identifies 10 ways in which computers can be used to promote learning:

1. Computer conducted drill and practice.
2. Computer assisted instruction.
3. Computer managed instruction.
4. Fully computerized instruction.
5. Use of computers to solve problems.
6. Use of computer and programming concepts to foster discovery and organization of ideas.
7. Learning in an information-technology-based environment that gives the student broad scope for initiative.
8. Practice in an information-technology-based environment that simulates a real-world task.
9. Learning by teaching human students in an information-technology-based environment.
10. Learning by teaching computers.

With the exception of number 3 above, computer managed instruction, students

24

and teachers at Upper St. Clair use these computer-enhanced learning modes through computer-based activities in standard mathematics and science courses, a computer science course for credit, independent study, their free periods, and before and after school hours. However, students with teacher-made assignments are given priority in using the terminals. For the most part, students do not learn programming as part of their coursework. Usually programming skills are learned at the computer-resource room with the aid of manuals, student assistants, and faculty who are available to answer questions. Programming topics come from diverse sources such as mathematics, science, and other classes; other students; ideas in magazines; and various extra-curricular and at-home activities. Before students are permitted to use the time-sharing terminals they first complete some prepared assignments using batch processing. Then they are permitted to use terminals to aid in doing class assignments, to pursue their own interests, and for constructive recreational activities. While game making, programming, and playing as independent learning activities are encouraged, casual time-killing game playing is not permitted.

The computer science independent study course for credit has specific requirements. Each student must write and run programs to order lists of words or numbers, solve equations, perform matrix operations, etc. Students, as an aid to algorithmic thinking and learning programming skills, learn to obtain and analyze traces of programs. Each student also completes a negotiated individual project to demonstrate his or her programming skills and creative abilities in using a computer. Such a project could be a game, simulation, or tutorial to be used by other students; it could be a graphics program to show various views of a building or piece of equipment. Since the computer center is run by two mathematics teachers and most non-mathematics teachers at Upper St. Clair are reluctant to become involved in using the computer center, in spite of encouragement and offers of assistance from the director of the center, many of the computer projects are mathematical in nature.

As part of the efforts to keep students, faculty, and parents informed of the computer-enhanced learning activities at Upper St. Clair and to promote computer literacy, James Saunders publishes a bimonthly newletter called *Computer News* that now circulates throughout Pennsylvania and in about 30 other states.

At Upper St. Clair, the computer-center, which is adjacent to the math/science resource center, is regarded as an open learning environment where students are free to pursue their own interests in a scholarly, research-oriented environment where knowledge, ability, and proficiency are more important status criteria than freshman, senior, or teacher labels. In the computer center students frequently assume the role of teacher and the teachers are willing to learn from the students.

All of the hardware, computer-enhanced courses, and other computer-based learning activities as well as the computer hardware, related resources, and space for the computer resource room have been obtained through the informed and persistent efforts of two mathematics teachers, Saunders and DeBlassio, in convincing the school administration and school board of the value in providing and continually improving a computer-based learning environment that is controlled by teachers and used exclusively by students and teachers. While 90 percent of the students graduating from Upper St. Clair have used a computer, the director of the computer center estimates that 75 percent of the graduates have experiences in using the computer system of sufficient quality and quantity that they can be considered computer literate. That is, they know what a computer is, how to use it in different ways, have learned a programming language, are aware of the capabilities and limitations of computers, and know about various applications of computers in society.

Computer Literacy at the University of Pittsburgh

Beginning in 1973, the Mathematics Education Program at the University of Pittsburgh required a computer-literacy/computer-uses-in-education course of undergraduates preparing to become secondary-school mathematics teachers and all mathematics education majors in master's or doctoral programs. In 1978 these two courses were opened as electives to secondary education majors in other subject areas. The undergraduate and graduate versions of these courses have similar general objectives, with advanced programming skills, more knowledge of the computer-enhanced-education literature, and a higher-level term project being required of students in the graduate course. The general requirements of both courses include:

1. the ability to program in BASIC at the level that this language is developed in *A Guided Tour of Computer Programming in BASIC* (Dwyer and Kaufmann, 1973).
2. a thorough understanding of the various modes in which computers are used in teaching and learning.
3. general computer literacy as comprehensive as that represented by

the cognitive and affective computer literacy objectives developed by the Special Products Division of the Minnesota Educational Computing Consortium (Johnson et. al., 1979).

4. development of a comprehensive computer-enhanced-learning module that could be used for one-week in a high school to teach a topic in some subject. (These modules average about 20 typewritten pages in length.)

Although the only required textbook for these courses is Dwyer and Kaufmann's book mentioned above, students are assigned readings from BASIC and the Personal Computer (Dwyer and Critchfield, 1978) and The Best of Creative Computing, Vols. I and II, edited by David Ahl. Selected readings are also assigned from journals such as:

1. AEDS Journal
2. Computers and People
3. Creative Computing
4. Journal of Computer Based Instruction
5. Technological Horizons in Education
6. The Computing Teacher
7. SIGCUE Bulletin
8. On Computing
9. Recreational Computing
10. Educational Technology

The following articles are representative of assigned readings:

By Frederick H. Bell --

1. Why is computer related learning so successful? Educational Technology, December, 1974.
2. What do kids think about computers? SIGCUE Bulletin, July, 1975.
3. Students and teachers: a perspective on computer enhanced learning. EDU Nineteen, DEC, Fall, 197
4. Classroom computers: beyond the 3 R's. Creative Computing, September, 1979.
5. Computer literacy, school mathematics, and problem solving: three for the price of one. AEDS Journal, Summer, 1979.

By Thomas A. Dwyer --

1. Computers and the romantic view of education. Technological Horizons in Education, No. 3, 1974.
2. Computers and the curriculum question. AEDS Journal, Spring, 1975.
3. The art of education: blueprint for a renaissance. Creative Computing, Sept-Oct, 1976.
4. Computers and teachers: getting our priorities straight. The Computing Teacher, Fall, 1979.

We are currently developing and evaluating a three-part computer literacy test on (1) programming and related skills, (2) general computer literacy knowledge, (3) knowledge and skills for computer uses in education. Representative items from each of these categories are:

1. What is the correct output for this program?

```
10 LET X$ = "*"
20 LET Y$ = "#"
30 FOR I = 1 TO 10
40 PRINT Y$; X$;
50 NEXT I
60 END
```

2. Most computers have the capability to:
   a. correct errors in the programs that control them.
   b. store both words and numbers in memory.
   c. think up ways to solve problems.
   d. all of a, b, and c.
   e. none of a, b, and c.
3. A serious factor that is discouraging the use of computers in schools is:
   a. lack of research about their effectiveness as an aid in teaching and learning.
   b. reluctance of many teachers and administrators to use computers.
   c. the high cost of computers.
   d. the negative side effects that computers have upon students' attitudes.
   e. the fact that computers are being used to replace teachers.

At the end of each undergraduate and graduate computer-literacy/computers-in-education course, students are asked to evaluate the course. Students' comments range from a negative, "I don't really like to write programs and probably will never work in a school where computers are used in the courses I teach." to a positive, "I really enjoy using computers and intend to take some more computer courses. Can you tell me which schools around Pittsburgh use computers so I can apply for jobs there?" The mode response tends to be similar to, "I'm glad I took this course and learned how to use computers and how they are used in society and schools." After taking one of the computer literacy courses, most students feel that they know "somewhat more" to "considerably more" than most people know about computers and are fairly confident of their ability to cope with a computer-oriented society.

In spite of the fact that these courses are required of secondary mathematics education majors but are electives for secondary education majors in other fields, approximately three-fourths of the students in the courses are not mathematics majors. Of approximately 100 students who have taken one or the other of these courses since 1978, the following major fields of study have been represented -- art, Latin, French, German, Spanish, English, special education, psychology, mathematics, biology, chemistry, physics, performing arts, history, and sociology. Although it would certainly be inappropriate to make any broad generalizations, English and mathematics majors have tended to be the best programmers and English and foreign language majors have tended to write the best computer-enhanced curriculum modules. Two generalizations that can be made are that students who tend to try to pass courses through memorizing content and imitating the way the professor solves problems have trouble learning programming skills. And students who are well organized and think in logical, orderly ways tend to be good programmers.

Many students begin their computer course with fear and trepidation -- fear of the unknown and apprehension about the hardware. Some people believe that they will have to learn how to operate a computer without wrecking it, much in the manner that one approaches learning to drive a car. In order to overcome these fears by showing that there is no basis for them, informality and handling a TRS-80 computer highlight the first few class sessions. The first class is devoted to demonstrating several calculators, ranging from the newest micro-technology to an old Monroe electric that clanks and bounces, a TRS-80 computer, and "Alphie," which is Playskool's foot-high, smiling, twenty-dollar "electronic robot," that plays tunes, plays games, answers questions, and flashes lights for children from "3-8 years" and older. "Alphie" takes the fear out of everyone, and after half an hour everyone is taking turns blowing up castles, making rectangle-art pictures, and composing poetry on the TRS-80 using pre-programmed tapes. In fact, most people who don't have another class next period stay around and play games for a while. After a few classes such as this, students are ready to go to an RJE station and learn how to LOGIN and LOGOUT on the Pitt computing system at Decwriter II terminals without fear, even though it is less fun than playing games on the TRS-80. The dropout rate in these elective courses is quite low -- about one student in twenty.

COOPERATION BETWEEN A SCHOOL DISTRICT AND UNIVERSITY

Although the developmental work, which is described in this paper, between the University of Pittsburgh and Upper St. Clair was the primary responsibility of Bell and Saunders and DeBlassio, respectively, its sucess was due in part to cooperative efforts between the school district and university. Through Bell, the University of Pittsburgh provided in-service education on educational uses of computers to teachers at Upper St. Clair and assisted in the development of computer-based curriculum materials and evaluation of the school district program. Conversely, both Saunders and DeBlassio, as a consequence of sabbatical leaves that were spent at Pitt, helped to develop and evaluate the computer courses that are offered to Secondary Education majors at the university. These two teachers earned doctorates in Mathematics Education with concentrations in educational applications of computers under the direction of Bell. Probably much of the succes of this cooperative project was due to informal discussions, correspondence, and exchange of ideas and information among the individuals involved.

In 1976, the University of Pittsburgh Computer Science and Mathematics Education Programs nominated the Upper St. Clair School District as an exemplary institution using computers for learning and teaching. The district was subsequently listed in the Academic Computer Directory of exemplary institutions using computers in education, which, under the direction of Seidel and Hunter (1977), was partially funded by the National Science Foundation. Upper St. Clair High School was cited as a school whose computer system is totally devoted to student use where students learn about computers and their uses through interaction with other students, teachers, and resources.

RESEARCH FINDINGS OF THE PROJECT

Between September 1977 and May 1978, two carefully controlled research studies were carried out on students who were using computers at Upper St. Clair High School. In the first study we explored the feasibility of integrating computer literacy in an algebra II course, where students did computer-based algebra assignments as part of the course. The questions that were addressed were:

1. To what extent can students become computer literate through using a computer in an algebra II course?
2. If computer topics are included in an algebra II course, will fewer algebra topics be covered and will student achievement levels drop?
3. How much extra time will a computer-enhanced algebra II course require of students and teachers?
4. Will the use of computers have a significant effect upon students' attitudes toward mathematics, the instructional setting, or computers?

In this study, 101 students were randomly assigned to four Algebra II classes.

Two of these classes used a computer throughout the course; the other two did not. Two teachers worked as a team in teaching the same algebra topics to all four classes on approximately the same days. However, students in the experimental computer group supplemented their study of algebra by doing weekly computer assignments — either using library programs to solve problems and explore topics or writing their own programs. One class period was used in orienting the experimental classes to the computer-resource room and showing them how to use the computer system. After that, students learned selected programming skills with the assistance of other students in the computer resource room and through independent study.

The following answers were found to the four questions stated above:

1. Most students were able to learn some fundamental BASIC programming skills as well as understand the capabilities and limitations of computers. They also learned how computers can be used as an aid in learning and problem solving.
2. Just as much material was covered in algebra II by students who used the computer, and no signficant differences were found between control and experimental classes' mean algebra II test scores.
3. Both students and teachers spent an average of less than one hour of extra time per week due to the computer topics in the experimental classes.
4. Analysis of students' scores on attitude scales showed that using the computer had no significant effect on students' attitudes toward mathematics and the instructional setting. However, a significant improvement in the experimental group's attitudes toward computers was observed between January and May 1978.

See Saunders (1978) for details of this study.

The second study (DeBlassio, 1978) was concerned with examining relationships between students' attitudes toward using a computer in a high school mathematics program and their attitudes toward mathematics and the instructional setting, mathematics achievement, programming achievement, intelligence quotient, and sex. We also wanted to see if there are certain attitude, ability, personality, and anxiety characteristics that are common to students who like to use a computer and if students who dislike using computers have common characteristics.

Data was gathered from 220 eleventh- and twelfth-grade students who had been using computers in their high-school mathematics classes. Major findings of this study are:

1. Statistically significant positive relationships were found between students' attitudes toward using a computer and (a) their attitudes toward the instructional setting, (b) their mathematics and programming achievement, and (c) their programming experience.
2. No significant relationships were found between students' attitudes toward using a computer and their intelligence-quotient scores.
3. Boys and girls had similar attitudes toward using a computer, and the two sexes did equally well in their computer-related mathematics classes.
4. The 52 students who liked using computers said they enjoyed the creative problem-solving aspect of writing programs.
5. The 28 students who disliked using computers were anxious due to the lack of structure and teacher supervision in the computer-resource room. The remaining 140 students had neutral attitudes toward computers.

Random samples of nine students from each of the groups who liked, disliked, or were neutral about using a computer were interviewed and given a battery of personality tests. People who liked using a computer tended to have:

1. favorable attitudes toward the instructional setting.
2. high mathematics achievement scores.
3. high programming ability.
4. a liking for problem solving.
5. above average vigor.
6. above average intelligence levels.

Students who disliked using computers had the following characteristics:

1. poor attitudes toward the instructional setting.
2. average ability in mathematics.
3. average programming ability.
4. neutral attitudes toward problem solving.
5. anxious, cautious, responsible, and non-vigorous personality traits.

In summary, our research over the past ten years provides strong support for integrating computer literacy and computer-based learning activities into high school mathematics programs. Students who used computers in mathematics classes tended to like mathematics and problem solving and to enjoy writing computer programs. While becoming computer literate as part of their mathematics program, students were able to learn the same number of mathematics topics with no negative effects upon their mathematics achievement.

REFERENCES

Ahl, David. The best of creative computing, vols. I and II. Morristown, New Jersey: Creative Computing.

Bell, Frederick H. Computer literacy, school mathematics and problem solving: three for the price of one. AEDS Journal, 1979, 12(4): 163-170.

Charp, S. and Altschuler, H. 1966-1976 a decade of usage of the computer for instructional purposes. Technological Horizons in Education, 1976, 3(8): 10-11,30.

DeBlassio, John. K. *Student characteristics associated with attitudes toward using a computer in selected high school mathematics courses.* Unpublished doctoral dissertation, University of Pittsburgh, 1978.

Dwyer, Thomas A. and Kaufmann, Michael S. *A guided tour of computer programming in BASIC.* Boston: Houghton Mifflin, 1973.

Dwyer, Thomas and Critchfield, Margot. *BASIC and the personal computer.* Reading, Massachusetts, 1978.

Johnson, David C., et. al. Computer literacy -- what is it? Minnesota Educational Computing Consortium, St. Paul Minnesota, 1979.

Licklider, J.C.R. Impact of information technology on education in science and technology. *Technology in Science and Education: The Next Ten Years.* Washington, D.C.: National Science Foundation, July 1979.

Molnar, Andrew. The use of computers in education. *Technological Horizons in Education*, 1976, 3(2): 18-23.

Molnar, Andrew. National policy toward technological innovation and academic computing. *Technological Horizons in Education*, 1977, 4(6): 39-43.

Saunders, James. What are the real problems involved in getting computers into the high school? *The Mathematics Teacher*, 1978, 71(5): 443-47.

Saunders, James. *The effects of using computer-enhanced resource materials on achievement and attitudes in second-year algebra.* Unpublished doctoral dissertation, University of Pittsburgh, 1978.

Saunders, James and DeBlassio, John. The computer goes to high school. *Proceedings of the Digital Equipment Users Society*, Chicago, Ill.: April 1978.

Seidel, Robert J. and Hunter, Beverly. *Academic computing directory.* Alexandria, Virginia: Human Resources Research Organization, 1977.

A COMPUTERIZED SYSTEM FOR PROJECTING STUDENT MEMBERSHIP IN SCHOOL SYSTEMS EXPERIENCING DRAMATIC CHANGES IN THEIR STUDENT ASSIGNMENT PLAN

Alvin M. Best & Neil G. Pedersen
Richmond Public Schools
Richmond, Virginia

The capability to accurately project student membership in public school systems plays a vital role in educational planning. The traditional cohort-survival method works adequately if the changes occuring in school districts are constant and homogeneous, but sweeping and radical changes which require redrawing school boundaries, closing schools, establishing magnet schools, and changing feeder systems make the traditional method all but unworkable. The Richmond, Virginia Public Schools are now basing their projections on student populations in geographic regions rather than membership in schools. The two keys to the system were the development of a geographically-based file which pinpoints students' addresses and the computerization and modification of the projection system.

Computer-Based Geographic Locations of Students.

During the 1978-79 school year a computerized system for the geographic location of students was developed (Rugg & Marriott, 1978). Before this time a hand spot mapping technique was used to locate and count pupils within a specified geographic area. Needless to say, there were major drawbacks to this procedure. Aside from the possibility of human error in a manual process and a lack of timeliness in the finished product, the procedure limited the demographic variables that could be considered (grade and race in our case). In addition, an attendance zone change could be very unwieldly. The manual spot mapping system was basically a static procedure with limited usefulness and narrow focus.

The crux of the computerized mapping system is the U. S. Census Bureau's DIME file (Dual Independent Map Encoding). Items such as streets, rivers, and school boundaries are all encoded in the DIME file used by the school system. By matching pupils' addresses with DIME street segments, the X and Y coordinates of every school child's house within the school boundaries can be interpolated. With a simple point-in-polygon routine the system determines which elementary, middle and high school boundaries are applicable to addresses and can assign census tract, block, and zip-code for future use. In addition to its use in the projections of student membership, the file may also be used to generate computerized maps using SYMAP (Dougenik & Sheehan, 1975) and other management information system reports.

Redefining a school boundary, closing or opening a school, and other such changes in the feeder system of schools thus no longer pose large problems. While changing the definition of the polygon that defines a school boundary is not a trivial problem, once it is done such a change allows those responsible for the dissemination of related information to produce any number of required and timely reports from already existing software. Reports for proposed programs and for student projections are thus developed more quickly and accurately.

Overview of the Enrollment Forecasting.

During the summer of 1979, a new approach to the projection of school enrollments was developed by a graduate student in the Department of Urban Studies and Planning at Virginia Commonwealth University (Crispell, 1979). A number of complex models were reviewed; the common survival ratio approach was investigated and rejected. Complex models taking into account population demographics were rejected because of the problems and uncertainties in the model and because it could not be shown to be more accurate than the survival ratio model. Bayesian models were studied in depth, but such an approach proved fruitless because of the inappropriateness of the model to the actual mechanics of the movement of pupils in the school system. For the range of projections needed in Richmond Public Schools, a simpler, more explanatory model was chosen. The present model generates projections for the system, each attendance zone, and each school separately for each grade level. The zone and school projections are done for one-year and, if necessary, for 5-years. The system projections for each grade level, with suitable assumptions, can probably be safely done for longer time periods in the future (a 10-year projection was completed).

System Enrollment Forecasting.

For system projections, the method uses previous year's grade enrollments to project the probable enrollments in future years' grades. A three-step model is used which incorporates the information on September and June membership in each grade and the promotions and retentions reported in June. These easily obtainable data are used to compute the rates defined below:

1. Winter Migration Rate

   The proportion of pupils who are in a grade at the end of the year (June) as compared to the number in that grade at the beginning of the year (September).

2. Promotion Rate

   The proportion of pupils in each grade who are promoted in June to a succeeding grade (the remainder are retained in that grade).

3. Summer Migration Rate

   The proportion of pupils who actually enroll in a grade in September who were promoted to or retained in that grade the previous June.

Except for enrichment classes, the projection model is quite simple. In order to forecast the number of pupils in a grade in successive years, two figures are added: the number of pupils who (a) "survive" the school year (winter migration) in the previous grade, (b) pass the previous grade, and (c) "survive" the summer (summer migration rate); and the number of pupils who (a) "survive" the school year in that grade, (b) are retained in that grade, and (c) "survive" the summer. Enrichment classes that do not follow the standard route, are handled slightly differently. For instance, Richmond has a pre-high class of eighth graders who are given special instruction for a year before moving to the 9th grade. As a result, eighth graders go one of three routes in the model: retained in the eighth grade, enrolled in Pre-High, or promoted to ninth grade. This is a simple modification to the above model.

Kindergarten must be handled slightly differently also because it has no grade feeding into it. Since 1970, about two-thirds of the number of children born in Richmond enroll in kindergarten five years later. Thus a proportion of the number of births (either actual or projected) is used to calculate projected kindergarten enrollment.

The system projections use overall grade totals, not the counts of students in each grade and attendance zone. However, the zone projection model used is nearly the same as the system projection model but the zone projections are done separately for each attendance zone. Since the system projections are believed to be more reliable, any zone or school projections are then adjusted to insure that the grade totals add up to system projection for that year. A particular zone's projection is simply scaled up or down to insure that the grand total for all grades in the zone projections are identical with the system projection for that grade.

Attendance Zone Forecasting.

In addition to the system projections, input to the geographically-based attendance zone projection system include (1) a count of all students in each grade residing in each elementary attendance zone in the city and (2) any migration or promotion rates which vary from zone to zone. The system promotion rates are used for the non-enrichment grades, but the system migration rates are not.

A migration rate is seen as a composite of the social and economic factors which affect the decline (or increase) in a grade during the school year and during the summer. Making reliable estimates of these effects for every attendance zone and grade is unlikely to be successful.

With the computerized student location system, it is possible to study any number of hypothesized trends that might be occuring differently in each zone and apply these migration or promotion rate differences to the model.

Any number of different sets of rate estimates can be experimented with to "fine-tune" or modify the model for future use, but the initial model used estimated only two migration rates for each attendance zone: a rate to be used for the elementary grades and another rate to be applied to the secondary grades. These zone migration rates were derived by using the present school boundaries on the past year's geographically-based student files. Such non-school effects on school membership can, therefore, be investigated using whatever school attendance zones are necessary.

Student records for 1976, 1977, and 1978 were conformed to the 1979 school year boundaries. The yearly elementary migration rates for each zone were derived by following the K-3 enrollment in 1976 through grades 1-4 in 1977 and 2-5 in 1978. The secondary migration rate was calculated in a similar fashion by following grades 6-10 in 1976. Finally, although 31 different sets of migration rates were calculated, two for each zone, the values were categorized into four values to be used in the actual projections. The result was a projection for each attendance zone and grade.

School Enrollment Forecasting.

These zone projections must then go through three stages to be converted to school projections. Firstly, the defined feeder system of the elementary attendance zones are applied to produce a school projection that could be used if no students attended schools out of their zone, if all schools followed the feeder system and, if no "personal judgment" was deemed necessary to modify the empirical projections.

The exceptions to the strict feeder system are again handled by the computerized student file. Students attending schools out-of zone, students attending schools which, in effect, have no attendance zones, and other exceptions are counted and applied to the raw school projections. One can, for instance, count the number of pupils attending the "wrong" school as defined by the feeder system, subtract them out of the assigned school, and add them into the other, more historically correct school. A so-called "grandfather clause" can be handled in this manner as well. It is assumed that students attending out-of-zone have valid reasons and will continue in their present school until they move up to the next feeder-system level.

The third stage is the application of personal adjustments to the school projections to reflect factors which cannot be empirically derived. All three of these stages simply involve the moving of pupils from one school to another within a grade; no grade totals are changed. One example of the third type of shifting of students would occur if a new, magnet school were opened and was designed to draw pupils from the whole city. The best that one could do might be to draw proportionately from each school to build the membership of the new school.

Discussion.

The development of a computerized, geographically-based student accounting file has made possible a number of developments in improved planning and management systems for the Richmond Public Schools. This paper provides an overview of its effect on the newly created enrollment projection system. The computerized file has made it possible to handle dramatic changes in the student assignment plan of the school system, to do analyses of the effects of such changes or the effects of on-going trends in the student population, and to develop a geographically-based projection system that may give more improved projections.

Although the present projections have not been empirically verified by next year's enrollments, application of the present system to the past year's data indicates that the system outlined in this paper should give more accurate projections than was the case in the past. Richmond Public Schools has found this a welcome enhancement to its planning procedures.

Rugg, R. D. & Marriott, D. R.
A case study of computer mapping system use in educational planning. Paper presented at the International User's Conference on Computer Mapping Software and Data Bases, Boston, Mass., July 25, 1978.

Dougenik, J. A. & Sheehan, D.
SYMAP User's Reference Manual
Cambridge, Mass.: Laboratory for Computer Graphics and Spatial Analysis, Harvard University, 1975.

Crispell, B. A student enrollment forecasting technique for the Richmond Public Schools, Richmond, Virginia. Unpublished manuscript, September 4, 1979.

A MULTI-MEDIA CAI TERMINAL BASED UPON A MICROPROCESSOR WITH APPLICATIONS FOR THE HANDICAPPED

ANN BREBNER AND H. J. HALLWORTH

Faculty of Education Computer Applications Unit and

Department of Educational Psychology

The University of Calgary

The Need for a Multi-Media Terminal

The Computer Applications Unit in the University of Calgary Faculty of Education has been engaged in research and development relating to computer assisted instruction and computer managed instruction since 1969. Programs have been developed and used for a wide variety of subject areas and levels, from first grade arithmetic and language to university statistics. For some applications a hard-copy or CRT terminal has been very effective. For other applications a variety of input and output devices have been desirable, and sometimes essential.

The need for such specialized devices has been particularly obvious in the case of a CAI project with the developmentally handicapped (Hallworth & Brebner, 1979). Some developmentally handicapped students make good progress using a Teletype or a CRT terminal. However, some with physical handicaps, especially the cerebral palsied, are unable to use a standard keyboard. Partially sighted students are unable to see normal size characters produced by a hard-copy or a CRT terminal.

Moreover, it has long been accepted that developmentally handicapped students should have programs relating specifically to their own interests and needs, and aimed at the development of "social survival skills" (Clarke & Clarke, 1965). Arithmetic, for example, should be "social arithmetic" centered around money handling and shopping, with the courseware incorporating images of coins, bills, and articles to be purchased. Programs to teach prereading, reading, and social sight vocabulary obviously require both image projection and audio output.

The CAI project at the Calgary Vocational and Rehabilitation Research Institute was started in 1971 and has shown that, in order to provide instruction appropriate for the developmentally handicapped, a "CAI station" is needed which incorporates a variety of input and output devices. Experience with CAI in other environments has also indicated that such a station is advantageous to many learners. This is the case, for example, with first grade students learning counting and reading (Moont, 1978), and with university students taking nursing courses.

CAI Station Interfaces

The Computer Applications Unit has always used a dedicated general purpose time-shared computer for its research and development work. However, since the software supplied with such machines has traditionally been designed to service only a standard terminal at each port, the use of a CAI station has necessitated an interface between the computer and the station.

The first such interfaces were designed by the Unit in 1969 to allow operation of both a terminal and a slide projector, and remained in use until 1979. An interface developed by the National Research Council was also used, which incorporated a CRT terminal, a slide projector, a tape recorder, and an ultra-sonic touch sensitive display (Hlady, 1969). A later commercial version of this design included a cassette recorder and a touch sensitive display with light emitting diodes.

However, although each was a significant advance, none of these interfaces provided an adequate solution to all problems. A satisfactory interface must be portable, in order to take CAI to the learner in his customary learning environment; it must be reliable, in order to promote credibility in the system; and it must be flexible, in order to take advantage of new input and output devices as they are required and become available.

Consequently, it was decided to design an interface based on a microprocessor. Fifteen such devices have now been built and have been in use for eighteen months.

A Microprocessor-Based Terminal

In order to obtain a reliable interface it was decided to use a microprocessor which had a proven record and for which there was ample software available. For this reason, the 8080 chip was selected. In order to ensure flexibility, it was necessary to have a standard bus for which a large number of manufacturers were already providing equipment, therefore the S-100 bus was used. Since the terminal was designed to be modular, with no large components, it could be easily transported.

The precise configuration of a terminal is determined by its use. The S-100 bus has space for up to 21 boards. Besides the CPU board the terminal may have, as memory, from 4K to 56K bytes of RAM and from 4K to 16K bytes of EPROM. A character generator board provides for 24 lines of 80 standard size characters, or 12 lines of 40 double size characters for use by the partially sighted; it also provides simple graphic characters. A graphics board gives control of 240 x 256 individual points on the CRT monitor, and could be used for special character sets, animation and, with the addition of two further boards, colour. A synthetic speech board is also in use, with digitized words stored either in EPROMs or on a floppy disk. Boards have been designed and constructed to control a random access slide projector, and to make a front panel unnecessary so that a terminal is activated simply by switching on the power. A multi-channel input-output board and an interrupt board are used and boards to control floppy disks are introduced when required.

The remaining equipment for the terminal includes a 17" CRT monitor, a random access slide projector, a rear view screen, and a speaker.

The software for the terminal consists of utility programs developed in 8080 code in order to facilitate the use of the boards referred to above. One such program services a light pen.

CAI programs are written for the host computer in DEC BASIC-PLUS, employing user defined functions for input and output and for control of the various station devices. For example, the function parameters to show a slide are "SLI" and the slide number; to output a message via the speech synthesizer they are "SAY" and the message number.

Each BASIC-PLUS program has attached to it a sub-routine which acts as a link to the utility programs.

The terminal keyboard has an EIA connection. This will accept a number pad, a light pen, or any of the special input devices presently in use or under development.

Applications with the Handicapped

The microprocessor-based terminal has proved invaluable for the CAI project with developmentally handicapped students. Not only has it been very reliable in operation but also its flexibility allows the courseware programmer to use a wide range of input and output devices.

In the case of the social arithmetic programs, for example, the standard form of output is by large size characters on a 17" CRT terminal. For partially sighted students, it was previously necessary to display standard size characters on a 23" video monitor. Slides of coins, bills, and articles to be purchased, can be used reliably with the terminal operating over a wide range of baud rates, whereas previous interfaces were restricted to 110 or 300 baud.

Special character sets may be designed and used on the CRT monitor. A possible future requirement for CAI with the handicapped, for example, is to display Bliss symbols.

Synthetic speech is already used to a limited degree, and can be expanded by the addition to the terminal of a floppy disk or other mass storage device. Although too many instructions confuse the handicapped (Brown & Hughson, 1972), a small amount of audio output has been demonstrated to enhance their learning (Holz, 1976).

As indicated earlier, some of the input devices presently in use are designed to overcome students' physical handicaps. These include, for example, the POSSUM apparatus. This consists of a transparent indicator board divided into a matrix having nine rows and columns, with a character in each cell. A light behind the board is moved first in a horizontal and then in a vertical plane, and is controlled by any binary switching device such as a joystick, a dual rod control, a footswitch, a handswitch. a headswitch, or a pneumatic mouthpiece on which the user sucks or puffs. Another special input device is the POSSUM Expanded Keyboard, a rugged box on which the student can lean whilst finding large and well spaced recessed keys. Cerebral palsied students have been able to make use of both pieces of equipment.

Other input devices are aimed less at overcoming physical handicaps than at facilitating learning by eliminating some of the superfluous mental processes otherwise required of the learner using the standard keyboard. In effect, a light pen and a touch sensitive display operate in this way: the learner answers a question merely by selecting it on the screen. Without such a device it would be necessary to code the answer either numerically or alphabetically and then enter the code via the standard keyboard.

However, a special keyboard may be designed to simplify the coding required. For example, a "money keyboard" has been developed for use with the social arithmetic programs. The keys are actual bills and coins. To enter an amount of money, the student merely touches the necessary "money" keys. For a program to teach the meaning of laundry symbols, a special keyboard is used with the actual symbols on the keys. A program to teach a sense of time uses keys which represent "Early", "On Time" and "Late". Basic nutrition is taught by a program which uses a keyboard with four keys representing the four food groups; and a program to train cafeteria cashiers uses an actual cash register as a keyboard by replacing the original keys by pieces of bubble-pad keyboard.

Obviously, the use of "concept" keys on a special keyboard, as described above, may be extended much further, and not only for use with the handicapped. The procedure has already been used successfully to teach statistics to university students (Taerum, 1979).

Present Developments

The microprocessor-based terminal has made CAI an effective and economic tool for teachers of developmentally handicapped students who are close to the host computer. Communications costs to rural areas, however, are prohibitive. The terminal has therefore been developed into a stand-alone computer by the addition of more RAM and dual floppy disks. This system has operated successfully for a period of nine months in a small town 80 miles from Calgary.

As the incorporation into the terminal of a videodisc system presents few technical problems, a project has been initiated to make appropriate developments to the existing courseware.

Mass storage used as a shared resource will hopefully reduce the cost of program distribution and record keeping. It will then be economically feasible to establish a multi-terminal system to supply CAI in any centre for the handicapped, whether urban or rural.

REFERENCES

Brown, R. I. and Hughson, E. A. Verbal instruction in the performance of non-verbal tasks. Report to the Alberta Human Resources Research Council, 1972.

Clarke, A. M. & Clarke, E. D. B. Mental Deficiency: The Changing Outlook. London: Methuen, 1965.

Hallworth, H. J. and Brebner, Ann. Development of computer assisted instruction at the VRRI. In R. I. Brown & M. B. Bayer (Eds.), Research, Demonstration, Practice: Ten Years of Progress. Calgary, Alberta: Vocational and Rehabilitation Research Institute, 1979.

Hlady, A. M. A touch sensitive X-Y position encoder for computer input. Proceedings of the Joint Fall Computer Conference, American Federation of Information Processing Societies, 1969.

Holz, Else. A study of the use of computer assisted instruction for teaching a social sight vocabulary to mentally handicapped adolescents. Unpublished M.Sc. thesis, University of Calgary, 1976.

Moont, Susan E. A Study of the use of computer assisted instruction for teaching a counting skill and a vocabulary skill at the Grade One level. Unpublished M.Sc. thesis, University of Calgary, 1978.

Taerum, T. V. A teaching procedure for computer assisted instruction in mathematics and statistics. Unpublished doctoral thesis, University of Calgary, 1979.

TEACHING ELEMENTARY READING BY CMI AND CAI

ANN BREBNER, H. J. HALLWORTH, ESTHER McINTOSH AND CECIL WONTNER

Faculty of Education Computer Applications Unit and

Department of Educational Psychology

The University of Calgary

Introduction

Various practices in use in the educational system today, such as promotion by age rather than ability and the integration of the handicapped into the normal schools, bring together groups of students with widely differing abilities and achievement levels. Teachers faced with educating such heterogeneous classes must introduce some small group or individualized instruction. While there is little opposition to the idea of such instruction it is rarely carried out on a wide scale due to the size of the practical difficulties involved.

There are two major problems. The first of these is to gather and arrange in a logical sequence sufficient varied learning materials on which students can work individually. The second problem relates to individualizing the use of these materials and includes assessing each student's strengths and weaknesses, prescribing appropriate learning materials and monitoring the progress of each student in the class. The latter problem, however, can be greatly alleviated by the use of a computer management system which performs routine testing and record keeping tasks.

The Reading Materials

Several years ago a committee was formed in Belvedere-Parkway Elementary School in Calgary to attempt to develop an organized group approach to individualized instruction. Priority was given to the area of reading.

Many sets of materials and different approaches to the teaching of reading skills were considered. It soon became obvious, however, that it was necessary first to specify a continuum of skills that was sequential from Kindergarten to Grade 8. Further, it was realized that some of the programs available needed far larger amounts of resource materials than were practically manageable in the school.

After several months of investigation it was decided that materials developed by the Educational Development Corporation (1973) most closely met the school's requirements. These materials related to a continuum of skills, defined in behavioral terms, and were compatible with the Calgary Board of Education's listings. They included a diagnostic testing program, a prescription service, and teacher support materials. Further, many of the latter were already available in Belvedere-Parkway School.

Having obtained the necessary learning packages, the problem of how to bring them into use was addressed.

Since an essential part of a reading skills program is continuous monitoring, pacing and evaluation with testing, diagnosis and prescription occurring throughout the entire school year, it was apparent from the beginning that an efficient management system was needed. Hence the present research project, namely a computer managed instruction program in reading, was initiated by a team from the Computer Applications Unit of the University of Calgary's Faculty of Education, the Calgary Board of Education Research and Testing Department, and the Belvedere-Parkway School committee.

The criterion referenced tests provided by the Educational Development Corp. are designed for Kindergarten to Grade 8, and consist of 329 behavioral objectives which are tested in 44 booklets. Each booklet contains two questions for each of the objectives, and most have 16 questions, thus testing 8 objectives. There are two sets of tests, one of which can be used as a pre-test and the other as a post-test, giving each student four attempts at each objective. This provides the teacher with a performance picture of the individual student.

The Computer Management System

A computer management system was written in the BASIC-PLUS language for the DEC 11/70 computer operated by the Computer Applications Unit. It incorporates three main file structures. The first contains the answers to the questions in the criterion referenced tests, another has a record for each student in the school, and the third file holds the prescriptions. Two smaller associated files contain lists of teacher names and objectives. The system is accessed by terminals in the reading area of the school using acoustic couplers and the standard telephone network.

The day-to-day running of the management system is carried out by an aide, whose duties include entering all student names into the student file at the beginning of the year, registering those who join during the year and deleting the records of those who leave, monitoring the testing of students, and obtaining reports for the two teachers responsible for the reading area.

Testing is usually done on-line. When the student's name is entered, the system accesses the appropriate record on the student file and prints out the number of the next test booklet which the student should complete. In the case of students working on the system for the first time, the booklet number is estimated by a reading teacher. Also a supplementary program is available to these students which will guide them through a series of steps to ensure they fully understand the testing procedure.

The student then reads the multiple choice questions from the booklet and types the answers on the terminal keyboard. The program scores these replies and stores the results for further analysis. Testing continues until a twenty-minute session has been completed on the terminal, or until a choice is made not to proceed further, or until the "approximate working level" of the student is reached. This level is defined as the last correct response beyond which the student gives five successive incorrect answers.

At this point the student summary report (Figure 1) may be obtained. It lists the objectives mastered (both answers correct), those which need to be reviewed (one answer correct), and those which must be learned (both answers wrong). These student summaries provide very useful teacher-student conference material.

When testing sessions have been completed for all the students a class summary (Figure 2) is available. This groups the students by objective into three sets. First those students who know the objective are listed, then the students who need to review the objective, and finally those who need to learn the objective.

For each of these three sets of students the class summary also lists a variety of prescriptions. This is a most important part of the management system in that it indicates in detail the precise parts of the reading instructional materials to which each student should now proceed. All prescriptions refer to texts used and available in the school. For every objective a prescription exists for three sets of teaching materials thus allowing for teacher or student choice, and also ensuring that supplementary material will always be available.

A final report available is a school summary which lists the reading performance of the students by objective. This performance rating indicates first the number of students who need to learn, then those who need to review, and finally those who know the objective. The report provides a reading performance level by grade and objective for the entire school and is used by the administration and library services.

Results

The management system has now operated for more than five years with a minimum number of technical problems and has been very well received by teachers, students and parents.

For the teacher there are a number of benefits from the program. Such routine tasks as administering and scoring tests, recording results, assessing, prescribing according to this assessment, grouping the students for instruction, and re-testing as the need arises are all carried out automatically by the computer management system. A further important benefit is that the prescriptions are available and grouping can be done when the test results are still relevant, rather than some time afterwards when the results are already becoming outdated. Teachers are not told how to teach the children, but are given individual assessments and groupings which may be used in conjunction with their own informal assessments and observations. They are therefore relieved of many clerical tasks, and so gain pupil contact hours.

The teachers find an additional advantage over and above the use of the computer printouts in the classroom, namely, in reporting student progress to parents. The printouts provide an excellent basis for both interviews and report cards.

Corrective reading teachers continually update reading prescriptions, replacing those which are not meeting the student or teacher expectations. A series of back-up remedial materials for all the objectives has been developed and is maintained by the corrective reading team. These materials are readily available to all teachers in the school.

All students see their own student summary and invariably accept the printout as an accurate statement of their skills. Moreover, they always receive at the top of the summary a positive statement of the form: "You have a good understanding of the following objectives". In over five years of operation, only one student has ever failed to receive this statement and, with it, quite positive encouragement. This motivates students to read the report further, to look at their needs and to improve their reading skills. This motivation had not been anticipated, and is an incidental but invaluable benefit from the program in developing students' reading skills.

The parents are particularly interested in seeing the report produced by the management system. From the computer printout they now know at reporting time exactly where their child is in relation to what is expected in that year's reading program.

One way in which the success of the program can be measured is by examing reading test results. Since the program was implemented these have improved considerably. On city-wide tests the mean performance of Belvedere-Parkway School Grade 6 students used to be at the 25th percentile. It has gradually increased until it is now at the 55th percentile.

Although these achievement results are important, the attitude change brought about by the program is more significant. By directing the student's attention to those items that require review and need to be learned, the individual reports save time and energy. Moreover, the recognition of the objectives that the student understands supplies a reason to use what has been learned, and to learn more. This factor has increased library usage, and the quality of all reading-based work.

Students from Belvedere-Parkway School proceed to a local junior high school. Teachers from this school confirm that there has been a significant improvement in the attitudes towards school and learning among students from Belvedere-Parkway School. They now frequently send junior high school students for testing and prescription by the computer management program.

Other schools both within Calgary and in other areas of the Province are interested in the program. The reading teachers frequently demonstrate it to personnel from other school districts, while within the city it has been used by several schools on a trial basis.

Three years ago all 400 students in a neighbouring elementary school were tested by the Belvedere-Parkway reading staff, who provided a diagnosis for each student by means of individual student summaries, also class groupings and prescription listings.

The Calgary Children's Hospital School consider the system particularly suitable for their students, many of whom are handicapped and in need of individualized attention. An in-service workshop was held for the teachers in the hospital school, while teachers in another elementary school are currently learning how to use the system, and within a few months will be operating it for all their students.

Board members and teachers from the Calgary Separate School Board examined the management program in operation and were impressed with the capabilities. Having decided to try it on a small scale before actually implementing it in a school, they bused 13 students from a special education class to Belvedere-Parkway School for testing and assessment.

With class loads increasing and integration of handicapped children into regular school classes becoming a reality, teachers and administrators face not only heavy teaching assignments but a great diversity of student abilities within one class. Computer technology has much to offer in solving these problems: Belvedere-Parkway School has found, from over five years of experience, that computer managed instruction does lead directly to individualization which in turn leads to better achievement and improved attitudes towards learning.

Conclusions

Stemming from the success of the initial project various other computer based learning activities are either being implemented or are in the planning stage.

The reading teachers are examining the objectives and changing them to better meet the needs of the local students. They are also examining statistics on the test questions and replacing those that are ambiguous or do not test the objective adequately.

Computer assisted instruction modules are being developed, and some have already been tested and are in regular use in the school. These cover the topics of syllabication and roots and affixes (Wontner, 1978), and plural forms and possessives (McIntosh, 1978).

Also, since the number of terminals in the school was increased from three to five in September, 1979, it is now feasible to consider the running of a further computer management system for arithmetic skills. It is planned to implement such a system in the near future when a suitable set of objectives, tests and prescriptions has been identified.

In summary, a computer management system for reading skills has proved very successful for both students and teachers at elementary school level. It is being expanded to include computer assisted instruction modules, and a further management system is planned for the learning of arithmetic skills.

REFERENCES

Individualized Criterion Referenced Testing. Educational Development Corporation, Tulsa, Okla., 1973

Wontner, C. A Study of the Effect of Using Computer Assisted Instruction for Teaching Vocabulary Skills at the Grade Six Level. Unpublished M.Sc. thesis, University of Calgary, 1978

McIntosh, Esther. CAI Programs on Plurals. Unpublished paper, University of Calgary, 1978

FIGURE 1

Student: 1051 September 1979
HERMAN LUK

STUDENT SUMMARY

YOU HAVE A GOOD UNDERSTANDING OF THE FOLLOWING OBJECTIVES:

234 Select short vowel word to complete a sentence
238 Define an unfamiliar word by selecting a synonym
241 Identify a phrase to complete a sentence by using context clues
242 Identify a word to complete a sentence by using context clues
243 Identify the definition of a word with multiple meanings by using context clues
247 Select a phrase that answers questions about an unfamiliar passage
257 Identify a word completing a statement about similarity or difference between characters in a story
262 Identify the purpose of a passage: to persuade, to inform or to entertain
264 Classify phrases: describing actions, forming visual pictures, denoting sounds

YOU NEED TO REVIEW THESE OBJECTIVES:

236 Identify the root of an affixated word (un*know*ingly)
237 Define correctly an affixated word
244 Match a word with multiple meanings to all its definitions
245 Identify key words in skimming a passage
246 Identify a sentence that means same as given sentence but has different functional units
251 Identify features that do or do not relate to a class of things
253 Identify a sentence which rephrases the main idea of a paragraph
254 Identify the phrase which completes the implied cause-effect relationship
255 Identify implied statement best supported by information in a passage
256 Identify a conclusion supporting or not supporting details in a passage
258 Select the correct word or phrase to complete a sentence in a familiar passage
259 Identify the best description of an author's intent
265 Identify statements of fact and opinion

YOU NEED TO LEARN THESE OBJECTIVES:

235 Syllabicate a three-syllable word
239 Identify an antonym
240 Identify a word for completing a sentence by finding its definition within the sentence
248 Identify time passed in a passage containing clues about time
249 Identify the statement making a logical inference based on a story
250 Identify a sentence which does not support the main idea
252 Identify a sentence which rephrases the main idea of a passage
260 Identify a sentence that summarizes a paragraph
263 Identify the emotion described in a passage

FIGURE 2

Room 55 Test # 277-Obj# 256 September 1979

IDENTIFY A CONCLUSION SUPPORTING OR NOT SUPPORTING DETAILS IN A PASSAGE

BOTH ITEMS CORRECT	A.R.P.L.[1]	OPEN HIGHWAY[2]	[4].R.A.
	6-6B	6:83*124*153	IDC:Books D*E*

Allen Lisa
Brandt Angela
Gray Frederick
Ho Sally
Johnson Richard
Kelner Charles
Mann Susan
Mason Margaret
Patterson Brian
Sordi Wendy
Vallence Charles
Wong Chun-Chi

ONE ITEM CORRECT	A.R.P.L.	OPEN HIGHWAY	.R.A.
	5-9A	5:48	IDC:Books C*D*

Carnduff Emily
Lee Robert
Muir Ken
Warren Joan

NO ITEM CORRECT	A.R.P.L.	OPEN HIGHWAY	.R.A.
	5-9A2	5T:157/5:30*48	IDC:Books B*C

Chan Henry
Stotyn Dennis

1. A.R.P.L. = Audio Reading Progress Laboratory
2. Open Highway = Reading Text series
3. .R.A. = various reading activities listed in 4.
4. IDC = A Barnell Loft reading booklet on "drawing conclusions"

A COMPUTER-BASED MANAGEMENT PROGRAM FOR SELECTION OF APPLICANTS TO ALLIED HEALTH EDUCATIONAL PROGRAMS

Robert Blood
Michael Serena
Myron Goldberg
Miami-Dade Community College

INTRODUCTION

The Medical Center Campus of Miami-Dade Community College represents a unique approach in the teaching of health related professions at the community college level. It is the first instance where a separate campus of a multi-campus community college has been dedicated to meeting the educational needs of students interested in pursuing health care occupations.

The Medical Center Campus offers programs in fourteen health related occupations. As in all institutions that offer allied health each program receives many more applicants than there are available spaces for students. Limited instructional and clinical practice space and the high cost of training have forced these schools to establish admissions criteria for student selection. Traditionally the matching of student applicants against these selection criteria has been paper oriented and hand managed, requiring many hours of clerical, faculty, and administrative time which in most instances, could better be used in the delivery of instructional services.

The purpose of this project, then, was to design a computer assisted selection system which would introduce more efficiency by minimizing the clerical tasks of the selection process, provide for more effective use of faculty and administrative time and, where possible, provide comprehensive and easily accessible information. In light of these considerations the goals of the project were to:

1) provide a means for acknowledging receipt of a program application,

2) provide accurate information to the applicant regarding their status in the selection process,

3) compare each applicant against established selection criteria,

4) provide the means to notify applicants of acceptance, rejection or standby status and,

5) establish a data base for reporting and research purposes.

The program selected for developing and testing a computer-assisted selection process was the Associate Degree Nursing (ADN) program. The bases for selecting this program were two: (1) most number of applicants and (2) objective and quantifiable admission criteria. Of the fourteen programs at Medical Center Campus, the ADN program receives by far the largest number of applications. The number of applicants for the last six entering classes averaged 481 with a range of 293 to 712. Of these applicants, the average number selected was 117, with a range of 84 to 145. In that the ADN program admits two classes per school year, approximately 900 applications are processed in a single year. In regards to admissions criteria, the ADN program employes a hierarchical classification system in which applicants are categorized in terms of most qualified to least qualified. Each category of the system is defined by specific academic information, including number of credits earned or achievement level within certain non-nursing classes. Such information may be readily input to and stored in computer files.

The initial consideration for a more mechanized and efficient selection process was the creation of an off-line computer software system dedicated to the storage of applicant information. The establishment of such a stand-alone system would provide a means by which admissions data for each applicant could be easily retrieved in a desirable format at the time selection decisions were to be made. In addition, a permanent record of applicant data for future research would be available. However, for this type of off-line system, the massive amount of applicant data, including all demographic and academic information, needed to arrive at selection decisions would require manual input. Such a system would not make use of the relevant applicant information available from existing College data files. Thus, a considerable amount of personnel resources would still be needed to operate and maintain a system which would essentially replicate data already on file within the College. More importantly, such a system would in no way facilitate admission decisions or the provision of feedback to each applicant concerning his or her academic status in relation to the selection criteria of the ADN program.

Further efforts toward a solution led to the investigation of an existing computer based instructional management program known as Response System with Variable Prescriptions (RSVP). RSVP was developed at Miami-Dade Community College primarily to assist the instructional process of diverse student populations. Through RSVP, instructors are aided in the management of up to 15,000 students in a single course by maintaining a record for each student and providing individualized feedback to each student.

Though essentially a stand-alone system, RSVP was installed in Miami-Dade's main frame computer and thus provided an excellent opportunity to establish an interface with existing College student record files.

Because of the flexibility of RSVP and the feasibility of linking with existing College files, RSVP was chosen as the foundation upon which to build a computer assisted selection system.

RSVP AND MIAMI-DADE'S COURSE INTERFACE FILE (CIF)

The Response System with Variable Prescriptions was developed at Miami-Dade Community College, as one approach to individualized instruction and evaluation through the aid of computers. RSVP is capable of storing specified attributes for each student and, based on commands given to the system by the user, communicates via an individualized letter, specific feedback regarding student performance on a test or survey. In sum, RSVP is an adaptive system to which a student responds and for which a user designs written explanations, responses, or evaluations. In RSVP language, these written communications are referred to as prescriptions. Directed by these prescriptions, the student progresses through the learning process.

Further, RSVP is "content free" enabling users to design any number of instructional/evaluative/management applications. Its capabilities and applications are limited only by the imagination and resourcefulness of the user.[1]

Information related to a proposed project and those students participating in the project is maintained in what are referred to as "files." These files form the basis for RSVP functions which are executed by nine programs. Briefly, the initial four RSVP programs, RSVP01 through RSVP04, specifically deal with computer space allocation for the project and inputting specific attribute data for each student into appropriate RSVP files. Data may be stored in the form of alphabetic, integer, decimal or coded information (e.g., SEX CODE: MALE = 1; FEMALE = 2). RSVP05 and RSVP06 allow the user to score tests, calculate various measures of central tendency and variability, and print individualized prescriptions based upon student attributes and test performance. Lastly, the final three RSVP programs, RSVP07 through RSVP09 provide the user with detailed output regarding stored information including a listing of all student data input, test statistics, and class rosters. In addition, frequency counts of all requested data, with the exception of alphabetic and decimal information, may be generated.

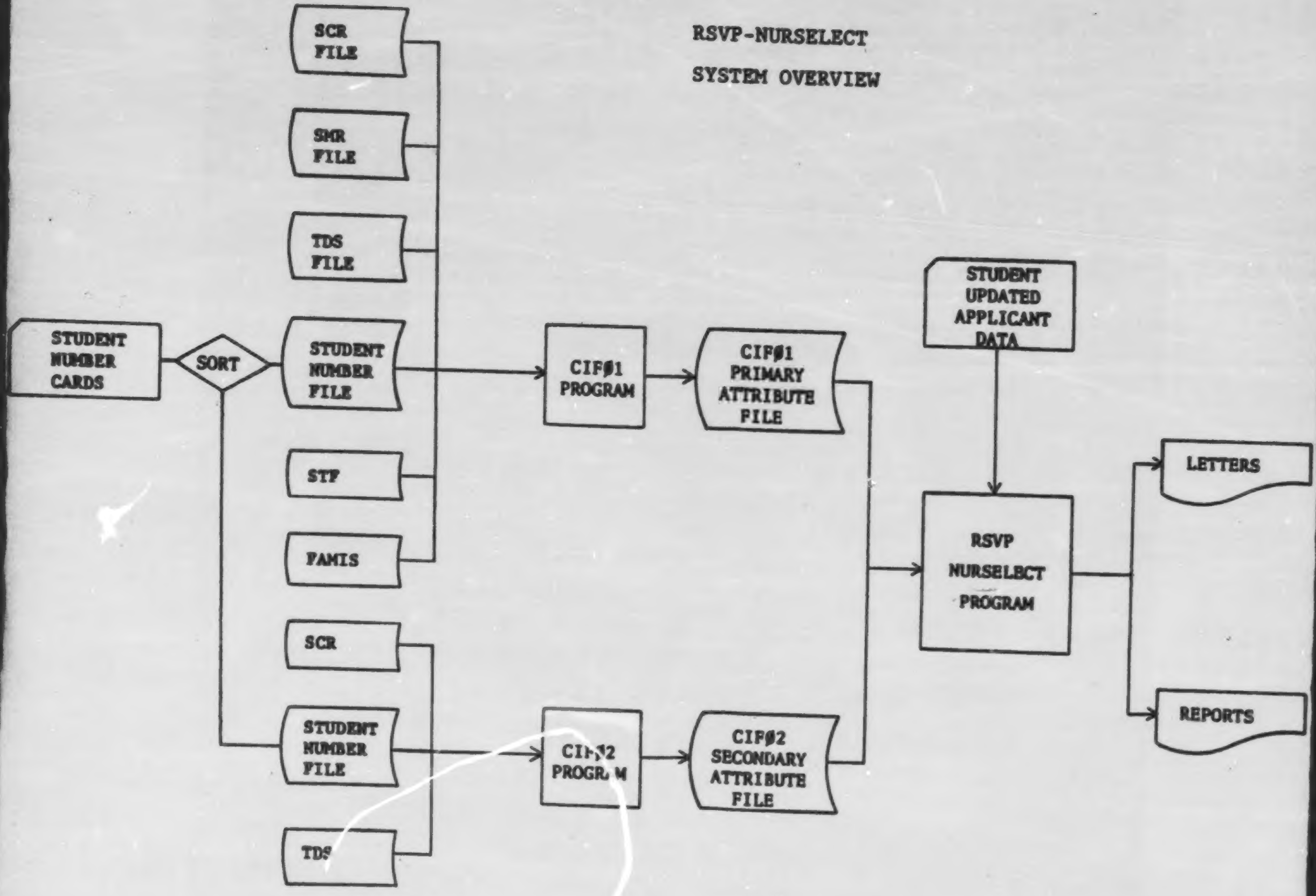
Currently, Miami-Dade Community College inputs all data regarding its student enrollment on five separate data bases; the Student Master Record (SMR) containing student demographic data, the Student Course Record (SCR) containing current semester course data, the College Student Transcript File (TDS) containing transcripts of all students who are actively enrolled, the Student Test File (STF) containing all standardized test results, and the Financial Aid Management Information System (FAMIS) containing information relating to Financial Aid recipients at Miami-Dade. As was mentioned earlier, RSVP has the capacity to store a wide variety of relevant student information. The process of inputting student data into an RSVP file was made less cumbersome and less time consuming with the creation of the Course Interface File (CIF). Based on user specification, CIF directly extracts from Miami-Dade's five data bases, which are updated daily, user requested information and creates a primary student attribute file which allows the input of data into any specified RSVP file. Once stored, RSVP can evaluate this data in terms of user specified criteria by way of decision rules. Based upon this evaluation, RSVP directs the printing of individualized prescriptions (letters) to each student.

RSVP-NURSELECT

The result of linking existing College files, through CIF with the file building and report writing capability of RSVP was RSVP-NURSELECT, a computer assisted system for efficient, individualized selection of students into the Associate Degree Nursing program at the Medical Center Campus of Miami-Dade Community College. A system overview of RSVP-NURSELECT is provided in Figure 1.

---

[1] For a more detailed explanaton of RSVP, the reader is referred to An Introduction to RSVP available through Auxiliary Services, Miami-Dade Community College, Miami, Florida 33176.



RSVP-NURSELECT

SYSTEM OVERVIEW

SCR FILE
SMR FILE
TDS FILE
STUDENT NUMBER CARDS
SORT
STUDENT NUMBER FILE
STF
FAMIS
SCR
STUDENT NUMBER FILE
TDS
CIF01 PROGRAM
CIF01 PRIMARY ATTRIBUTE FILE
CIF02 PROGRAM
CIF02 SECONDARY ATTRIBUTE FILE
STUDENT UPDATED APPLICANT DATA
RSVP NURSELECT PROGRAM
LETTERS
REPORTS

Figure 1

As can be seen in Figure 1, all information is keyed to the student identification number, a unique seven digit number assigned to the student, by the College admissions area, at the time the student is accepted to the College. For the purpose of the nursing selection process this number is supplied to the RSVP management area by the nursing department.

As the student number file is processed through the CIF01 program the specified record files are searched to determine if there is a record in that file for that particular student number. If there is a match the appropriate information is extracted from the file being referenced. As files are referenced the extracted information is used to build a primary attribute record for each student. This primary attribute file can be accessed directly by the RSVP system. It is important to note that the CIF01 program was designed to provide information regarding student attributes to any user of RSVP and is not limited to use in the NURSELECT system.

The CIF02 program was developed specifically for the student selection system. This program references the Student Transcript file and Student Course Record file and extracts data on those specific non-nursing courses which are considered in the selection criteria. CIF02 was also designed to reference nursing courses so that an academic profile can be maintained as a student, admitted to nursing, progresses through the curriculum.

CIF02 also operates on the basis of student identification number. From the Transcript file the program extracts the highest grade earned in the specified courses, calculates the number of times the course has been attempted, calculates the number of times the course has been dropped, and records the most recent year and term in which the course was taken. From the Student Course Record file the program extracts data as it relates to whether or not a student is currently registered for one of the specified courses.

A specific program in RSVP (RSVP04) is used to store all extracted data compiled by CIF01 and CIF02. This processed information represents the most updated profile of each applicant to the Associate Degree Nursing Program. It is this data in RSVP04 that is analyzed in comparison to the eighteen hierarchial criteria that determine which applicants are selected to the program before others. The eighteen criteria are ranked according to student overall grade point average with 2.0 or "C" as the minimum as well as the greatest number of sixteen non-nursing courses the student has completed with a grade of "C" or better required in science and a "D" or better in non-science. In addition to the above evaluation, each applicant must be seventeen years of age, must be a Florida resident, and must have graduated or possess a Graduate Equivalent Degree from high school.

The printing function provided by RSVP06 is paramount to the efficient operation of RSVP-NURSELECT. This program is responsible for the evaluation and selection of prospective applicants for admission into the Associate Degree Nursing program. Further, RSVP06 generates an individualized response (letter) to all students informing them of their application status. Selection of applicants and the individualized letters are generated through the use of previously established criteria (decision rules) which are based on each student's attribute and performance information accessed by CIF01 and CIF02 from the SMR, SCR, TDS, STF, and FAMIS files and stored in RSVP04.

As a result each applicant to the Associate Degree Nursing program receives one of the several possible versions of the following feedback letters:

1) acceptance into the program due to percentile score of 62 or above on the standardized test SCAT (School and College Ability Test),

2) rejection due to a percentile score of 61 or below on the SCAT,

3) rejection due to a failure to meet initial Associate Degree Nursing program prerequisites (17 years of age, Florida resident, and high school graduate or GED),

4) request for the student to send additional data in order to complete the application process,

5) acceptance into the ADN program by virtue of qualifying for one of the eighteen prespecified criteria,

6) rejection due to not qualifying for one of the eighteen prespecified criteria,

7) rejection due to the fact that the maximum number of applicants has been achieved, or

8) informing the student that he is being placed on "standby" in order to accommodate for any "no shows."

In summation, the implementation of RSVP-NURSELECT provides an extremely effective method for managing many of the essential tasks related to the applicant selection process of the Associate Degree Nursing program. The utility of this computer assisted selection system lies in its capacity to: (1) access and merge student information originating from existing on-line computer files and data card input; (2) perform decisions concerning applicants' selection status with respect to established acceptance, standby, and rejection criteria; (3) generate and print individualized letters; and (4) store and retrieve academic and demographic information on all applicants for future research and reports. Among several desirable effects, the application of this system results in a reduction in the amount of time and personnel support (clerical, faculty and administrative) needed to complete the nursing applicant selection process. In addition, the system affords the capability of providing each applicant, on a case-by-case basis, selection status feedback and individualized prescriptions for maintaining or improving academic standing within the College.

In the future, the potential for extensive academic research will be made possible by the data storage and retrieval ability of RSVP-NURSELECT. Two likely directions for this research are, (1) the evaluation of selection criteria used in admissions and, (2) determination of factors which relate to successful and unsuccessful academic performance. Moreover, in light of the success of RSVP-NURSELECT, plans are currently underway to apply the techniques developed in this project to the selection processes used within other allied health educational programs at Medical Center Campus.

INTERACTION OF COGNITIVE STYLE WITH TYPE OF FEEDBACK USED IN COMPUTER-ASSISTED EQUATION SOLVING

Vicki A. Boysen
Rex A. Thomas

Iowa State University

INTRODUCTION

The instructional usage of computers has undergone many and varied improvements since the concept was first introduced in the 1950's. Computers are now programmed to tutor, provide drill and practice, and simulate the real world. Computers also provide corrective feedback as they teach, usually by showing the correct answer when a question has been answered incorrectly. However, the benefit of feedback in computer-assisted instruction is still unproven despite its popular use and appeal.

> A computer-assisted instruction (CAI) system can provide detailed, immediate feedback following every student response. This is a theoretical advantage, but whether it is a real one remains to be seen. To be sure, one of the best documented propositions in psychology is that feedback facilitates learning.... But experimental demonstrations of the value of immediate feedback using actual lessons have been rare (Tait, Hartley, and Anderson, 1973; p. 161).

Renzi (1974) reviewed the literature on the effects of feedback on achievement and found that results were mixed. In some instances feedback was helpful or had no effect, while in others feedback even seemed to hinder achievement. Similar reviews by Ausubel and Robinson (1969) and Anderson, Kulhavy, and Andre (1971) reached the same conclusion. It appeared to researchers that these conflicting results might be accounted for by the presence of another variable interacting with feedback.

One variable which may account for this interaction is cognitive style. Kogan (1971) defines cognitive style as "individual variation in modes of perceiving, remembering, and thinking, or as distinctive ways of apprehending, storing, transforming, and utilizing information" (p. 244). Kogan emphasizes the word "modes" because cognitive style is concerned with the form rather than the content of cognitive activity. The definition thus describes a process rather than a product.

There are many different types of cognitive styles; Kogan (1971) defines nine. However, one of the best known and best researched is field dependence-independence, first described by Witkin and his associates (Witkin et al., 1954; Witkin et al., 1962; Witkin et al., 1977). Field dependence-independence is assessed by measuring an individual's perceptual ability to overcome an embedding context, or the extent to which he or she perceives part of an organized field as discrete from the surrounding field. People whose perception is strongly dominated by the surrounding field are designated "field dependent," while those who are able to separate a critical element from its surrounding field are labeled "field independent." These designations are not absolute. Rather, they represent extremes of a continuum along which lie scores representing tendencies, in varying degrees of strength, toward one mode of perception or the other.

Witkin et al. (1977) has theorized that, in problem-solving situations, field dependent individuals have difficulty taking a critical element out of the context in which it is presented and restructuring the problem material so that the critical element is used in a different context. He suggested that field dependent individuals may therefore need to be provided with more assistance in identifying critical elements in unstructured problems, while field independent individuals may need less assistance. Several studies in the area of mathematics education have tended to confirm this theory. A significant interaction between field dependence-independence and level of instructional guidance was found in at least three of these studies (Renzi, 1974; Bien, 1974; McLeod et al., 1978), and similar non-significant interactions were found in two others (McLeod and Adams, 1979; Adams and McLeod, 1979). The studies by Renzi (1974) and Adams and McLeod (1979) dealt directly with the effect of feedback, as a form of guidance, on achievement in mathematics and found that field dependent students receiving feedback achieved better than those not receiving feedback. Little difference in achievement, however, was found between field independent students receiving and not receiving feedback.

The present study investigated the interaction between field dependence-independence and type of feedback in a student-directed CAI lesson on algebraic equation solving. In addition, this study concurrently examined the possible interaction between feedback and a second aptitude variable -- mathematics achievement -- as recommended by Cronbach and Snow (1977). Because previous studies occasionally have found no interaction between feedback (or guidance) and cognitive style on immediate retention tests but have found significant interactions between guidance and cognitive style on transfer tests, a measure of transfer is included in the study.

METHOD

The treatments used in the study were two computer programs developed by the authors to provide practice in elementary equation solving. Both programs were written in BASIC for the Commodore PET microcomputer, and both required users to solve simple linear equations by requesting the computer to perform operations on the equations. Acceptable requests were of the form "M2" or "A-5/8X". The former multiplied each side of the equation by two, and the latter added -5/8X to both sides. Once the request had been made, only one program provided feedback as to whether the requested operation was the correct way to solve the equation. This "explicit" feedback program performed only operations that would simplify the equation. It monitored each operation as it was requested and, if the one requested by the subject was wrong, displayed the correct operation.

The second program, on the other hand, performed any legal operation requested by the user regardless of whether the operation simplified the equation or complicated it. This program was designated as the "implicit" feedback treatment because it displayed the results of all equation operations but required users to judge for themselves whether these operations were correct. For example, if the equation 3X+6=9 were presented and the user specified "A6", the implicit program produced the equation 3X+12=15. In contrast, the explicit program simply told the user to add negative six.

There were six levels of equations in both programs, ranging in difficulty from "X+2=5" to "-3/2X + 7/8 = 2/3X - 1/7". The content and sequence of these levels were based on those found in standard algebra textbooks. Both programs had identical types of equations at each level. However, within each level the equations were randomly generated so that a specific equation was unlikely to appear more than once. There was a 60% probability that the denominator of each term would be +1, and a 20% probability that two- and three-term equations would be reversed (i.e., "4=X-1" rather than "X-1=4").

The subjects in the study included 96 eighth grade students (41 girls and 55 boys) enrolled in four sections of a general mathematics course taught by the same instructor. Only one person in all four sections chose not to volunteer for the study, and only one other person chose to drop out of the study once it had begun.

Before they received the treatments, all subjects were given instruction by their regular teacher in solving simple linear equations. Next, the Group Embedded Figures Test (Witkin et al., 1971) was administered to measure each subject's degree of field dependence-independence. Stratified random sampling based on results of the Group Embedded Figures Test (GEFT) was then used to assign subjects to the treatment groups. Once the treatment groups were formed, subjects were given training in how to use the computer and how to communicate with it in order to solve an equation.

During the four-week treatment period each subject used the computer on the average of one day in three, a total of six computer sessions. Each session was fifteen minutes long, covered one level of equations, and took place during class. The primary author supervised each subject's work with the computer, but only to answer questions about the computer and to assist in choosing the correct program and level. No help on solving equations was given, except to refer subjects to pages in the textbook where this help could be found.

After the treatments had been completed, an instructor-made posttest measuring achievement in equation solving was administered. Equations on the test had a range of difficulty identical to the six levels in the computer programs. In addition to showing the solution to each equation, subjects were required to state what operation they had performed at each step in the solution. Subjects' scores on the mathematics portion of the Stanford Achievement Test (Advanced Level) were then obtained from student records. Following the experiment, the subjects' regular instructor taught them how to solve simultaneous equations and later tested them on that topic at the end of the unit. This test was used as a measure of transfer.

The posttest on equation solving and the test on simultaneous equations were both evaluated in two steps. First, the equation operations specified by the subject were checked ("operations"). Credit was given for any legal operation as long as it did not add a term to either side of the equation. Second, the computational correctness of each operation was checked ("computations"). Non-productive operations received computation credit if the operations were carried out correctly. After this, the test on simultaneous equations was further split into two more parts -- the first part consisting of steps in the solution concerned strictly with simultaneous equations (adding, subtracting, and substituting to eliminate one variable), and the second of steps required to solve the resulting equation for the other variable. The second part of the test on simultaneous equations measured subjects' ability to transfer previously learned skills in equation solving to new kinds of situations requiring this skill. This part of the test will hereafter be referred to as the Equation Solving Transfer Test.

RESULTS

The experimental plan of the study was an aptitude-treatment interaction (ATI) design using multiple regression analysis (Cronbach and Snow, 1977). The full regression model contained three main factors -- program (implicit or explicit feedback), GEFT score, and SAT mathematics score -- plus the interactions of program by GEFT and program by SAT. These interactions were the central focus of the study.

Table 1 presents the descriptive data for the study. Included are the means and standard deviations of all tests for both treatment groups. No significant differences were found between the groups on any of the tests.

The data in Table 2 however, indicate that significant interactions ($p<.05$) did occur on the operations and computations measures of the test of equation solving transfer. Both interactions involve the Group Embedded Figures test. Other results not listed in the table revealed that the factor of SAT mathematics scores was highly significant ($p<.001$) in accounting for subjects' scores on all

TABLE 1

Means and standard deviations of all tests

		Treatment Groups					
	Max	Implicit Feedback			Explicit Feedback		
Test	Score	Mean	SD	N	Mean	SD	N
GEFT	18	10.35	4.55	48	9.96	4.59	48
SAT Math	120	80.15	17.77	46	77.60	18.82	43
Posttest							
Operations[1]	1.0	.72	.22	48	.72	.24	48
Computations[1]	1.0	.52	.27	48	.50	.30	48
Simultaneous Eq							
Operations[1]	1.0	.79	.25	46	.77	.27	44
Computations[1]	1.0	.71	.26	46	.67	.27	44
Eq Solv Transfer							
Operations[1]	1.0	.73	.27	46	.70	.28	44
Computations[1]	1.0	.66	.28	46	.65	.27	44

[1] Raw scores transformed to proportions.

TABLE 2

Tests for interaction

Dependent Variable	R2 for full model	Source	F
Posttest			
Operations	.56	Prog by GEFT	.45
		Prog by SAT Math	.19
Computations	.57	Prog by GEFT	.81
		Prog by SAT Math	.00
Simultaneous Eq			
Operations	.33	Prog by GEFT	1.28
		Prog by SAT Math	.50
Computations	.41	Prog by GEFT	1.53
		Prog by SAT Math	.98
Eq Solv Transfer			
Operations	.40	Prog by GEFT	5.14*
		Prog by SAT Math	1.64
Computations	.42	Prog by GEFT	5.76*
		Prog by SAT Math	2.57

* $p<.05$

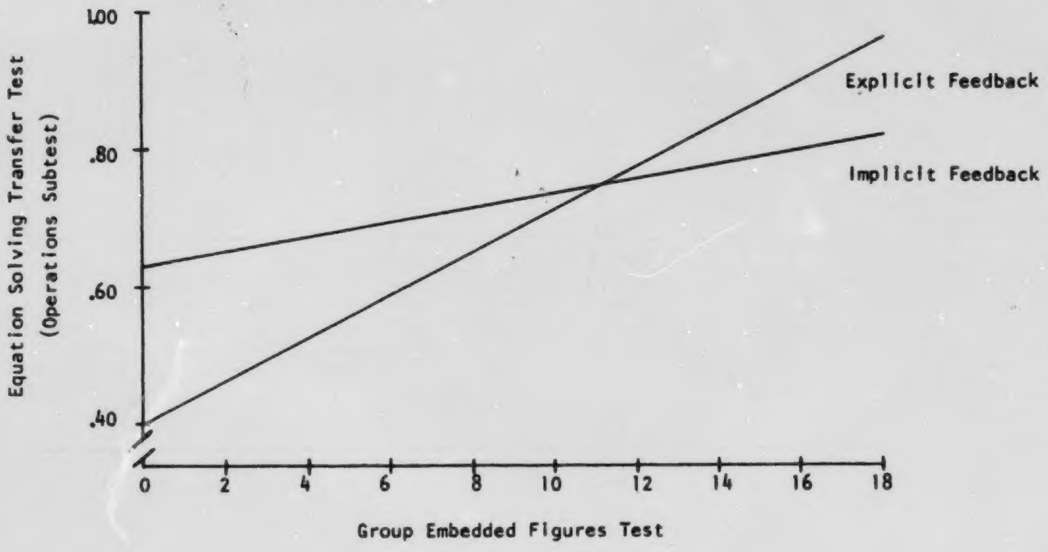


Figure 1: Regression of equation solving transfer test (operations subtest) on group embedded figures test.1

TABLE 3

Regression of posttest and transfer tests on the Group Embedded Figures Test.

Test	Treatment Group	GEFT Intercept	GEFT Regr. Coeff.
Posttest			
Operations	Implicit	.52	.02
	Explicit	.53	.02
Computations	Implicit	.31	.02
	Explicit	.26	.03
Simultaneous Eq			
Operations	Implicit	.66	.01
	Explicit	.55	.02
Computations	Implicit	.56	.01
	Explicit	.45	.02
Eq Solv Transfer			
Operations	Implicit	.63	.01
	Explicit	.41	.03
Computations	Implicit	.55	.01
	Explicit	.36	.03

dependent variables, while the factors of program and GEFT were not significant.

Figure 1 illustrates the interaction of GEFT with feedback on the operations subtest of the Equation Solving Transfer Test. The interaction of GEFT with feedback on the computations part of the same test was similar. In both cases the explicit feedback group had the steepest slope (.03), and the interactions were disordinal. The GEFT regression coefficients in Table 3 confirm the trend shown in Figure 1. On all tests except one there was a steeper slope for the explicit feedback program than for the implicit program.

DISCUSSION

The direction of the interaction found in this study was the opposite of that predicted by Witkin's theory of cognitive style. Instead of field dependent subjects achieving best under the high-structure explicit feedback program, they had higher test scores after using the low-structure implicit program. Conversely, field independent subjects performed better under the explicit rather than the implicit program. This interaction was statisticaly significant only for the two equation solving transfer test scores. However, the trend was consistent for all but one of the dependent variables.

The observed results might be attributed to the design of the two programs. the implicit program may have promoted an understanding of the process of equation solving by forcing the student to take the initiative. The explicit program, on the other hand, may have promoted memorization by providing maximum guidance. The field dependent subjects may have had only a vague understanding of the analytic concepts of equation solving at the time of the experimental treatment. If this was the case, they may not have been ready to take advantage of the drill provided by the explicit program. The implicit program, however, may have promoted their intuitive understanding by allowing them to solve equations in their own idiosyncratic ways.

In contrast, the field independent subjects may have had a clearer understanding of the concepts. In this case, the structured drill of the explicit program may have been a more timely and efficient learning aid than the unstructured implicit program.

In the present study, the treatments were administered after instruction and served as a means of practice. After examining the results of this study, however, it appears that the implicit feedback program might be more useful before instruction and the explicit feedback program after instruction. Used in this way, the implicit program would help the student develop an intuitive understanding of equation solving and prepare the student for classroom instruction. After classroom instruction, the explicit program could be used to reinforce the student's newly acquired understanding of the equation solving process. Further research is planned to investigate cognitive style and the use of feedback in both pre-instruction and post-instruction settings.

REFERENCES

Adams, V.M., & McLeod, D.B. The interaction of field dependence/independence and the level of guidance of mathematics instruction. Journal for Research in Mathematics Education, 1979, 10(5), 347-355.

Anderson, R.C., Kulhavy, R.W., & Andre, T. Feedback procedures in programmed instruction. Journal of Educational Psychology, 1971, 62(2), 148-156.

Ausubel, D.P., & Robinson, F.G. School learning: An introduction to educational psychology. New York: Holt, Rinehart, and Winston, 1969.

Bien, E.C. The relationship of cognitive style and structure of arithmetic materials to performance in fourth grade arithmetic. Unpublished doctoral dissertation. University of Pensylvania, 1974.

Cronbach, L.J., & Snow, R.E. Aptitudes and instructional methods: A handbook for research on interactions. New York: Irvington Publishers, 1977.

Kogan, N. Educational implications of cognitive styles. Pages 242-292 in G.S. Lesser, ed., Psychology and educational practice. Glenview, Ill.: Scott, Foresman, 1971.

McLeod, D.B., Carpenter, T.P., McCormack, R.L., & Skvarcius, R. Cognitive style and mathematics learning: The interaction of field independence and instructional treatment in numeration systems. Journal for Research in Mathematics Education, 1978, 9(3), 163-174.

McLeod, D.B., & Adams, V.M. Individual differences in cognitive style and discovery approaches to learning mathematics. The Journal of Educational Research, 1979, 72(6), 317-320.

Renzi, N.B. A study of some effects of field dependence-independence and feedback on performance achievement. Unpublished doctoral dissertation. Hofstra University, 1974.

Tait, K., Hartley, J.R., & Anderson, R.C. Feedback procedures in computer-assisted arithmetic instruction. The British Journal of Educational Psychology, 1973, 43(2), 161-171.

Witkin, H.A., Lewis, H.B., Hertzman, M., Machover, K., Meissner, P.B., & Wapner, S. Personality through perception. New York: Harper & Brothers, 1954.

Witkin, H.A., Dyk, R.B., Faterson, H.F., Goodenough, D.R., & Karp, S.A. Psychological differentiation: Studies of development. New York: John Wiley and Sons, 1962.

Witkin, H.A., Oltman, P.K., Raskin, E., & Karp, S.A. Manual for the Group Embedded Figures Test. Palo Alto, Calif.: Consulting Psychologists Press, 1971.

Witkin, H.A., Moore, C.A., Goodenough, D.R., & Cox, P.W. Field-dependent and field-independent cognitive styles and their educational implications. Review of Educational Research, 1977, 47(1), 1-64.

A DESIGN FOR THE EVALUATION OF MANAGEMENT INFORMATION SYSTEMS

William C. Bozeman
The University of Iowa

and

Dennis W. Spuck
The University of Houston

The role of management information systems as decision-support devices has received considerable attention in the literature, particularly with respect to design, implementation, operation, and technical considerations. An important dimension which has not been extensively examined concerns evaluation and models/procedures related to evaluation. This paper presents an evaluation schema consisting of (a) a prescriptive systems framework and (b) an evaluation framework. The first serves as a device for the identification of system elements while the latter permits focus on various areas of evaluation information. The procedures described consider qualitative as well as quantitative data and psychological/behavioral dynamics in the organization within a total systems approach.

Technology, as associated with information management and decision making, has received considerable attention in the literature in the past decade. This is due largely to the role of computers in business, industry, education and even our daily lives. Of course the concepts related to information management are not new and principles of information management can be found throughout recorded history. The advent of low-cost, high speed data processing facilities, however, has afforded the manager with resources which were not feasible only a few years ago. This availability and the increasing complexity of the management functions have provided the basis for the rapidly evolving field of management information systems.

Though the literature is replete with various definitions, discussions, and reviews of MIS, the focus of most writing is toward design, implementation, and operation of these systems. Much of that discussion is concerned with the technological aspects, including hardware/software, data base management, and other technical areas. Despite its importance, little attention has been devoted to the evaluation of a MIS.

The relative absence of the evaluation dimension in the MIS literature is somewhat ironic given the development of MIS technology since the early sixties. The initial phase of computer-based MIS in the early to mid-sixties was characterized by great enthusiasm and optimism. The second phase, from the late sixties to the early seventies, was marked by considerable disappointment. The present phase is characterized by cautious reappraisal (Berlet, 1977). The evolutionary/revolutionary development of the electronic data processing systems which made possible the sophisticated and elegant technology employed in MIS perhaps overshadowed other critical phases of the system life-cycle, such as evaluation. Clearly, this component must now receive the attention it demands.

PLANNING AND DESIGN OF AN MIS EVALUATION SYSTEM

Evaluation of MIS can assume many forms due to different purposes or functions which the evaluation effort is to address. These differences are a function of the various personnel roles in the organization, the perceived needs for the MIS, different levels of personal involvement, varying personality types, and levels of technical expertise. For example, the systems analyst is concerned with measures of effectiveness related to the functional operation of the system; the programmer or programming team seeks evaluative data regarding the efficiency of code, algorithms, and technical processes; systems engineers/analysts and programmers require evaluation of hardware/software components; operators are concerned with ease of use. The manager is often unconcerned with the technical aspects but is interested in the utility, appropriateness, accuracy, reliability, and timeliness of reports; policy makers and strategic planners will question long-range benefits and applicability to strategic decision making.

These functions of an evaluation system will vary with time, spanning the total MIS life cycle from initial planning to actual implementation and operation. The needs and purposes of evaluation at different points within the system life cycle lead to the so-called formative/summative evaluation frameworks and strategies. Formative evaluation generally refers to a process assessment and analysis of the extent to which progress is being made toward the achievement of the objectives for which the system was designed. Summative evaluation more directly relates to the impact and effects of the MIS on the organizational functions.

There can exist no single evaluation model or strategy which will be applicable to all MIS or

52

completely transportable from system to system. It is possible, however, to employ a generalized procedure which can be adapted to many, if not most, MIS evaluations. This procedure/model embodies four important principles:

1. A prescriptive definition of the term system.
2. The meaningful involvement of persons within the organization who have an interest in the MIS.
3. Consideration of both objective/quantifiable evaluation data and subjective/qualitative information.
4. A program of continuing change and improvement of the MIS.

These four principles are addressed in the following sections.

SYSTEM DEFINITION

In order to achieve the first principle, a system framework is employed as a foundation for defining the various components of the MIS. The criteria on which this framework is based are descriptiveness, completeness, and understandability by the user. The system framework must also be capable of depicting the MIS from a holistic perspective, capturing the basic design and operational characteristics. Such a model, in a sense, can then provide a "checklist" for the various MIS components.

Though there are numerous concepts and definitions of a system, the framework which appears most appropriate for the evaluation scheme presented in this paper is the system matrix conceived by Nadler (1979). This framework includes eight elements described in six dimensions.

The eight elements carry familiar and fairly well-understood names: (a) purpose (mission, function, or aim); examples of purpose are to provide credit information, to control flow of material resources, to manage manpower resources, to provide marketing intelligence information, or to support instructional decision making; (b) inputs (physical, human, and/or information), items on which conversion or processing takes place to arrive at outputs; inputs could be customer records, account data, inventory, personnel records, survey data, or student scores; (c) outputs (desired and undesired conversion of inputs); examples are credit reports, production control, facility design, personnel assignments, distribution models, or instructional recommendations; (d) sequence (order of operation or activities; sequence may involve parallel or series sub-systems of events including human activity as well as machine processing; (e) environment (physical and sociological setting); physical factors may include space and power requirements, noise levels, etc., while sociological factors may include the social system, organizational structure, roles, control, and morale; (f) human agents (human resources that aid in the processing sequence); examples are computer staff, systems analysts, and researchers; (g) physical catalysts (physical resources required in the processing sequence); these resources may include computer hardware, various media, buildings, office equipment, etc.; (h) information aids (knowledge and information resources used in the sequence events without becoming parts of the output); operating manuals, documentation, instructions, policies, and/or procedures may be information aids.

Specification of dimensions, as properties of these eight elements, recognizes that additional characterization of the elements is needed. The six dimensions are: (a) fundamental (observable properties or physical characteristics); (b) values (societal organizational desires, beliefs, or concensus; (c) measures (effectiveness measures, rates, capacities, performance, etc.); (d) control (methods used for measuring what occurs and specification performance levels, confidence limits, or degree of precision); (e) interface (relationships of system elements with other elements and/or operations in the organization); (f) future (planned changes for the element). The entire system framework can be viewed as an eighy-by-six grid with 48 cells.

INVOLVEMENT OF PEOPLE

The operationalization and utilization of this framework represents a means of achieving the second principle -- meaningful involvement of persons within the organization who have an interest in the MIS and are concerned with the overall MIS evaluation effort. The recommended strategy is the use of project teams. These groups can be composed of persons concerned with specific aspects of the system. For example, three evaluation teams might be designated: (a) a group composed of persons with technical expertness (e.g., programmers, analysts, engineers, etc.); (b) users of the MIS (e.g., managers, policy and decision makers); (c) operators and other persons concerned with certain quasi-technical operations in the system. Another alternative would be to form a team or teams which draw together individuals from each of these groups. Regardless of the alternative selected, the objective is to ensure that concerned persons with differing perspectives are involved in system evaluation as they similarly would have been in system design and implementation. The respective groups may operate under the direction of a central committee or person(s) charged with the overall responsibility for the evaluation. The initial phase of the model involves the analysis of the system by the project teams using the system framework.

EVALUATION: SYSTEMS, ELEMENTS AND DIMENSIONS

Each of the eight elements in the system matrix provides direction in the formulation of an evaluation design. System purpose is of overriding concern and together with system output points to system effects, that which is to be accomplished through system operation.

In MIS, the output is information, while the purpose of the system may be to improve decision making and ultimately to improve the effectiveness and/or efficiency of the system. Output may be the first focus of system evaluation when a determination is made as to (a) whether the system can function to produce desired outputs (reports containing information); (b) to what extent that output is used, for any purpose; and (c) the extent to which the use of that information affects organization decision making and productivity. As referenced in the description of outputs, both intended and unintended effects are of interest in system evaluation.

The steps or sequence to be taken to produce outputs identify points at which interim assessments of system functioning and operation may be made. System processing or sequence involves the interaction of physical catalysts and human agents within an environment. The physical catalysts are equipment and facilities which are used in the processing activities. Design specifications for these catalysts provide initial assessment criteria, but during system implementation the function of these catalysts may also be examined to determine whether they are imposing restrictions which constrain system effects. It is possible that the physical catalysts' requirements have been over specified and too great a performance capacity has been included, which may increase operational costs.

Human agents interact in the system in a variety of ways and the capability of these agents to perform properly is central to system effectiveness. Organization personnel may be called upon to provide input, carry out steps in system processing, or to utilize provided information in management functions. In the same way that specifications may be set for hardware functioning, specifications may be set for the required level of capability and performance of human agents. Pre-service training or recruitment programs are initiated prior to system operation to enable personnel to carry out roles required for system operation. In most cases, the attainment of required performance levels may be assessed prior to involvement in system operations, as well as on the job during system implementation. Training programs may be easier to develop and results easier to assess for some functions than for others -- likely it would be easier to train an individual to code data on mark-sense input forms than to train the individual to use reported information for improved decision making. In most training programs and in system operation, information aids such as training materials or reference books, may play a central role. These aids, too, should be a focus of system evaluation.

The system environment refers to the physical and attitudinal characteristics of the setting and points to factors both external and internal which may facilitate or hinder system effectiveness. With user attitude having been identified as a critical component in the success of computer-based management information systems, the monitoring and reporting of changes in this area may be especially vital. Training programs should be designed to take account of affective as well as performance and knowledge objectives.

The raw material or input of a management information system is data. Without data to aggregate and report as required for decision making, the system cannot function. While sources of data and data capturing mechanisms are addressed during planning, it is important to insure during the early stages of system evaluation that all required data elements can be accounted for, and that the data provided is reliable, valid, and updated as necessary. If processing steps are carried out with inaccurate data, it may not be until decisions based upon this data are implemented that the invalidity of the data is identified. Timeliness of updating is a related concern. Mechanisms may be in place to provide and update data elements, but unless this process is carried out in concert with decision-making needs based on this data, system purposes will not be fulfilled. Preliminary assessment of data-information sequencing may become apparent through system analysis and simulation, prior to implementation. Remaining problems will be detected during system implementation.

The six dimensions of the systems' framework also provide direction for the evaluation of management information systems. Certainly, the measures, control and fundamental dimensions reside at the very core of the evaluation process. The measures dimension can provide information of immediate and obvious need in the evaluation; in a sense, the elements of this dimension may constitute inputs to the evaluation system. The control dimension specifies how this information can be generated as an on-going and integral part of the MIS. The critical and fundamental nature of these dimensions cannot be over-stated as they provide a linkage to the evaluative mechanisms. Values set boundaries on acceptable practice and need to be considered as reference points for the analysis of behavior or expected behavior within the system. If actual or expected behavior is contrary to existing norms or beliefs, members of the organization may react negatively to the system, perhaps withhold cooperative behaviors and in some cases work to defeat the success of the system. For example, if systems implementation leads to the elimination of some positions or if the system contains individual productivity information, seen as being illegitimate, workers may see the system negatively and withhold participation or reduce cooperation.

There is a possibility when one utilizes a matrix such as the system framework, which separates the system into component parts, to overlook the integration of the parts when planning or evaluating the system. The interface dimension considers this integration. A problem may result, for example, not from a problem with the equipment or with the training of personnel, but from the inability of personnel to use equipment to carry out a given task.

The remaining dimension, future, leads us to consider the purposes of the evaluation effort itself. It provides information useful in system refinement and enhancement, as well as to allow for a decision considering system viability.

In the next section, the factors identified through the analysis of the elements and dimensions of the systems framework are incorporated into an evaluation model for management information systems.

EVALUATION FRAMEWORK

Primary purposes for the evaluation of a management information system are to assess the extent to which the goals and objectives of the system are being realized and to identify factors associated with successful and unsuccessful outcomes. Accomplishment of these purposes can and should lead to change and improvement of the MIS. Building upon the systems framework presented in the previous section, it is possible to identify the key elements associated with system success. Focus on those elements during system evaluation may provide information useful in refining system design and implementation, thereby increasing the likelihood of system effectiveness when implementation is completed.

To assist in the evaluation of management information systems, an evaluative framework has been developed. This framework includes three dimensions of evaluation and three levels of types of information. Evaluative dimensions include functional, utilization, and effect. The functional dimension is concerned with whether or not the various subsystems of the management information system (human, hardware and software) are capable of operating or functioning in accordance with design expectations. It is assumed that design specifications reflect actual need and that these specifications are sufficiently explicit to allow for assessments of system functioning to be carried out.

System utilization is concerned with an analysis of those management processes for which the system is being employed, whether or not this use is consistent with those specified in the system's design. Usage different from that specified is not immediately interpretable as a problem, as users frequently find creative and productive new uses of systems. Lack of use always constitutes a problem, the roots of which typically reside in inadequate system design or planning for implementation, the criteria used as the basis for the evaluation of system functioning. System utilization is dependent upon system functioning and system effectiveness is dependent upon both functioning and utilization. Assessment of functioning and utilization, thus, provides important intermediate criteria for the evaluation of system effectiveness.

System effects should directly reflect system purposes and goals. What results are being achieved as a result of system utilization and are the objectives of the systems being met? For example, in educational institutions goals ultimately include reference to changes in student behavior or student achievement. While objectives pertaining to changes in student behavior should not be ignored in the evaluation of management information systems, it frequently is difficult to establish a direct linkage between MIS implementation and changes in student variables. Factors affected more directly by system implementation or judgments concerning such variables might be investigated. For example changes in the way teachers or administrators use time, or judgments concerning the system's value in improving decision making could be assessed.

The distinction between directly observable system effects and user descriptions of system effects leads to the second dimension of the evaluation model -- the type of information available for system assessment. Information type is classified as actual, perceptual and attitudinal. Actual information is objective, derived from data which are generally observable and quantifiable.

Perceptual information is derived from user descriptions or perceptions of system operations and effects. Assessment of direct system effects on student achievement or system efficiency, thus, constitutes actual information while descriptions of perceived changes in student behavior or usefulness of information are perceptual information. Generally, actual information is better for evaluating system functioning, utilization and effect, but comparisons between actual and perceptual information can be of value in identifying problems in system operation, and in some cases the collection of actual information may be more difficult or costly than perceptual information.

The last category of information is attitudinal. Attitudinal information, comprised of user conclusions or judgments about the value and benefit of the system may include personal and intuitive feelings of the user toward the system (e.g., expressions of user apprehension, anxiety and satisfaction).

The three dimensions of evaluation and three levels of information when considered together comprise a nine-cell matrix.

An illustration of the framework applied to the evaluation of an instructional management information system or computer managed instruction system is given in Figure 1. The purpose of this system is to provide information which will improve instructional decision making, leading to the implementation of instructional programs which are individualized and maximize the attainment of the objectives in the instructional program being managed. A complete description of the evaluation is provided by Spuck, Bozeman, and Lawrence (1977).

CONCLUSION

This paper has presented a model for the evaluation of management information systems. Three levels of information considered were actual, perceptual and attitudinal and the dimensions of evaluation discussed were functional, utilization and effects. The three dimensions of evaluation are seen as hierarchical, in that utilization is dependent upon functionality and positive effect is dependent upon utilization. The functional dimension is concerned with whether the various subsystems of the MIS are capable of operating in accordance with design specifications. Utilization focuses on the management processes for which the MIS is being used and the effects dimension

DIMENSIONS / LEVELS	FUNCTIONAL Are the various subsystems of the MIS program, both human physical, capable of operating or functioning in accordance with design specifications?	UTILIZATION For what management processes is the system being employed and is the use of the system consistent with those uses specified in the design?	EFFECTS What results are achieved from the utilization of the system, and are the objectives of the system being met?
ACTUAL Objective and/or quantifiable information from primary source	1. Computer software testing 2. System response time 3. Required hardware available 4. Appropriate user interfaces (e.g. error messages) 5. User knowledge of system operations 6. System availability 7. User documentation and training materials	1. Number and type of system accesses 2. Uses of system reports 3. System responsiveness	1. Student achievement 2. Teacher time usage 3. System cost and cost effectiveness 4. Changes in roles and tasks
PERCEPTUAL User descriptions and awareness of system operations and effects	1. Need for MIS as perceived by user 2. Perceived level of administrative support	1. Descriptions of usefulness of system reports 2. Appropriateness of reports 3. System responsiveness	1. Descriptions of impact on achievement 2. Perceptions of system impact on decision making
ATTITUDINAL User conclusions about value of benefit of the system	1. Initial apprehension or anxiety 2. Potential utility 3. Personal commitment	1. Confidence in system accuracy and reliability 2. Feelings concerning system usability	1. Desireability of system effects 2. User satisfaction

Figure 1. Evaluation Framework for a CMI System

includes the results which are achieved through system use, whether intended or unintended. While the evaluation framework presented is more general than its application to MIS, the model has tailored to this application through an example in instructional management information systems.

Evaluation is an on-going process, conducted in conjunction with systems needs assessment, planning, design and implementation. As an integral part of these processes, it provides information which will support system modification to meet better the needs of the organization and its members. The need for system modification may result, for example, from changes in personnel, internal reorganization, changes in organizational goals or from variance between what is planned and what is able to be implemented. Formative evaluation strategies provide the information base to support adaptive changes, as well as longer term decisions concerning the value of the system, its effects and whether or not it should be continued or expanded.

REFERENCES

Berlet, H. Integrated data processing and management information systems: An overview. In the Diebold Group (Eds.), Automatic data processing handbook. New York: McGraw-Hill, 1977.

Nadler, G. The planning and design professions: An operational theory. Book in preparation, 1979.

Spuck, D.W., Bozeman, W.C., & Lawrence, B.F. Evaluation of the Wisconsin system for instructional management (Technical Report No. 438). Madison: Wisconsin Research and Development Center for Cognitive Learning, 1977.

DATABASE PROCESSING: TEACHING VIA AN INSTRUCTIONAL DATA BASE SYSTEM

Norman D. Brammer, Ph.D., CDP
Associate Professor
Management Science and Information Systems
College of Business
Colorado State University
Fort Collins, Colorado 80523

ABSTRACT: Based on the high level of interest in Data Base Management Systems within the Data Processing industry and the potential of such systems to improve the processing and presentation of information to managers and to improve the general function of data management, a curriculum offering in Data Base Processing has become highly important and relevant for those students looking toward careers in information systems. The important topical coverage for such a course will include (1) Data Structures, (2) Data Base Models, (3) Data Base Administration, (4) Design of Data Base Systems, and (5) Using a Data Base System.

Based on the belief that the real success of such a course is dependent on being able to actually utilize a functioning Data Base system rather than simply rely on text readings and lecture presentations, the use of an Instructional Data Base System as well as other Data Base Management Systems available at Colorado State University in a project environment will be considered. The use of such systems is a vital part of the course as they will enhance and reinforce the students' knowledge and appreciation of Data Base Processing Systems as well as increase their understanding of the topics listed above. The Instructional Data Base System (a data base management system developed at Colorado State) is available to other universities.

INTRODUCTION

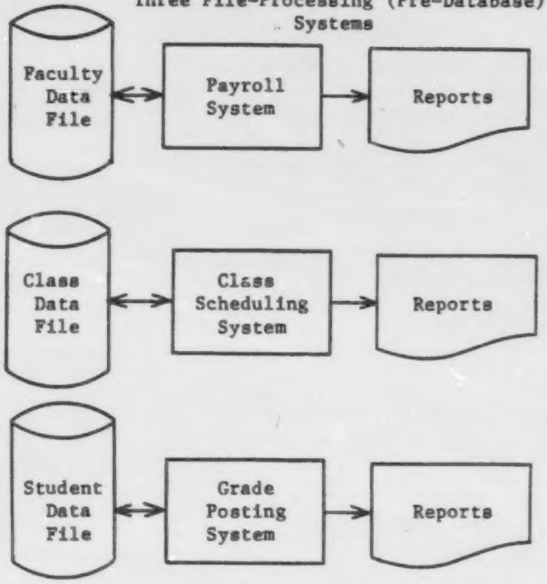
Database processing is currently a hot topic. Organizations are converting to or installing database systems in large numbers. Professional associations such as the Association for Computing Machinery (ACM) and Data Processing Management Association (DPMA) are sponsoring many conferences and seminars on database subjects. Recent issues of Datamation, Computer World, Computer Decisions, Infosystems and similar publications contain many database articles. Why?

Database processing seems to meet a critical need within organizations seeking improved control of their data and better information for the decision-maker. Certainly, not every organization needs a database, and for those that do or have tried such implementations, not all such efforts have been successful. In spite of this, the interest in database systems continues to increase.

What is database processing? Basically, database technology allows an organization to process data as an integrated whole. It reduces the artificiality imposed by traditional file processing, i.e., separate files for separate applications, and permits users to access data in a manner more natural to them.

To appreciate this concept, consider the file processing systems shown in Figure 1. They do not allow integrated processing: each processes its particular data, and they are effective in that they do produce checks, schedule classes, and record grades.

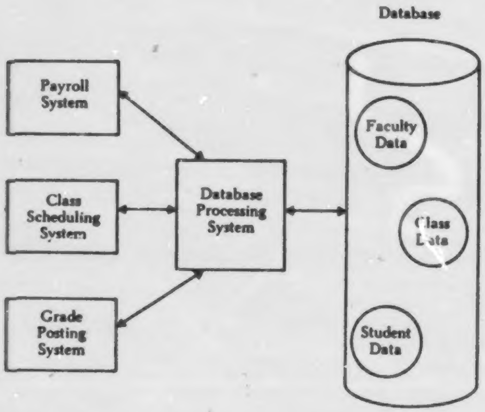
Figure 1
Three File-Processing (Pre-Database) Systems



Suppose someone wants to know the salary paid to each instructor who teaches a class scheduled by the class scheduling system. To obtain this information, a new program must be written to extract data from both files - faculty and class data. Unfortunately, there is no guarantee the files are compatible, particularly if they have been written in different languages. While it may be possible to satisfy such a request, it may take an inordinate amount of time and effort. Such a situation leads to the condition where we know information is "in the computer", but we can't get it out with reasonable effort.

Figure 2 shows a database processing system. The separate files from Figure 1 have been integrated into a database that is now processed indirectly by the applications, and hence can perform their old functions; but they call the database system to access the database. Now the data can be processed as an integrated whole as the files created by the database processing system are now compatible. For example, a faculty record can be logically "tied" to several class records to represent the relationship between teacher and class.

Figure 2
A Database System



The integration of the data offers several other advantages as well:

1. New requests and one-of-a-kind requests are more easily implemented.

2. Elimination of data duplication or redundancy.

3. Program/data independence.

4. Affordable sophisticated programming.

Of course, there are also disadvantages to database processing which can be summarized as:

1. Expensive
   Software
   More Hardware
1. Expensive
   Operating Costs

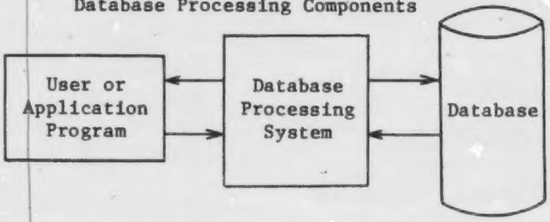
2. Complex

3. Recovery more difficult

4. Increased vulnerability to failure

Three basic components are involved in database processing: the user, the database processing system (usually referred to as the Data Base Management System), and the Database itself. The interaction and relationships of these components is shown in Figure 3.

Figure 3
Database Processing Components



DATA BASE COURSE CONTENT

Given the importance of database processing and therefore the need for a course in higher education curriculums, what is the topical coverage that is critical in such a course in order to adequately prepare students for entry into positions in organizations within the field of database processing?

Topical coverage for such a course in order to provide students a rich and rigorous background should cover:

1. Data Structures
2. Data Base Models
3. Data Base Administration
4. Design of Data Base Systems
5. Using a Data Base System

Data Structures: A concept central to understanding the current generation of data base management systems (DBMS) is that databases have structures by which data are stored and organized. These structures are as real in their own way as are architectural constructions, even though the support for them (as in the case of the building framework) are invisible to the user. Hence, understanding of database structures is basic to understanding DBMS but also for the ability to assess differences between available DBMS.

The following types of structures have relevance in database processing, and students should have a sound understanding of them:

1. Linked lists which are used to physically represent data relationships.

2. Hierarchical or tree logical structures.

3. Networks - simple or complex logical structures.

4. Inverted structures.

5. Relational representation.

All of the above structures are important to the understanding of representing data in databases and should be thoroughly covered in a course in database processing.

Data Base Models

The basic purpose of considering models is to provide a general description of systems which can be used to solve many different database processing problems and also to provide a general relationship to the structures above. Further, the models then can provide a framework and a means of evaluating and comparing the multitude of DBMS now available.

Three basic models can be used for this part of the course. They are:

1. Data Language/I or DL/I model (IBM) as a representation of the hierarchical structure/model.

2. CODASYL DBTG as a representation of network structure/model.

3. Relational model which is largely theoretical at this time, but which has great promise for future activity.

Generally, database models are made up of two parts, the Data Definition Language (DDL) for describing the structure of a database, and the Data Manipulation Language (DML) for describing the way the database is processed. Such coverage is therefore highly important for students and should bring on further appreciation of the general idea of structures and a general feeling for and appreciation of how a database can be processed by users.

Data Base Administration

One thing has been made clear based on organizations' experience with database processing, and that is the need for a position or office in an organization called the Data Base Administration (DBA). This is so because of the integration of data and that the data is shared by many users. It is important that students fully appreciate this position and its duties which include as its principal functions:

1. Database System Selection
2. Database Creation
3. Database Maintenance
4. Database Guardian

Full coverage of the above functions and their importance should be included in such a course.

Design of Data Base Systems

The implementation of a successful database system or project is more complex, time consuming, and costly than the implementation of any other application software system. The database project can involve every aspect of an organization. It requires the integration of data and the data processing activities of many diverse users. Successful database projects require careful planning followed by deliberate action. A database system cannot be just put in for users to play with.

On the other hand, the implementation of a database system does not require any revolutionary method of analysis and design. Techniques used for successful file-processing systems can be used for database systems. However, the techniques will apply to a broader range of activity, and the analysis and design will be more complex.

The steps involved in database analysis and design for a successful implementation should include:

1. Preliminary Analysis
2. Identify Objectives
3. Specify Alternatives
4. Evaluate Alternatives
5. Design and Implement System

The suggested steps are tried and true techniques for implementation of a successful system. They differ only in the magnitude and complexity when dealing with a database system as compared to a traditional file-processing application.

While students may be exposed to such steps in other course work, their importance should be carefully related and emphasized in the database environment.

Using a Database System

The actual use of a database system is a very important part of a course in database processing. The consideration of such a system and its importance will be seen later. In connection with using a database system, there are many other important concepts which should be developed. These concepts essentially enable a student to get a full grasp of the environment surrounding an operational database. Here, we will briefly consider the concepts of importance. They are:

1. The cast of characters involved with a database can be put into three groups. There are the users for which the database is designed and operated to satisfy their needs. There are database administrators who, in conjunction with the users, design, create, maintain and guard the database. Finally, there are system specialists, the liaison between the host computer system and the database. They are concerned with the physical structure of the database on the computer. The role, position and interaction of these groups is an important concept and should be fully appreciated by students.

2. The classification of data base users is important due to the widely different requirements of different users. To assist in the description of these requirements, there are five possible ways users can be classifed:

a. By the method used to direct the database system--whether he uses query language or a high level host programming language.

b. By the mode of access to the database--whether he batches inputs and processors periodically or whether he will process interactively--one transaction at a time.

c. By the nature of the user's requests against the database--are they of a fixed and specific nature or of a dynamic nature limited only by the query or host language used.

d. By the range of data referenced--the entire database or only a portion thereof.

e. By items to be accessed--either the data itself or the data structure.

The classification of users in such a manner then provides useful dimensions for planning and designing a database application.

3. The methods used by a user to identify data should become very familiar to students so that he questions how a database system implements them. They are:

a. Sequential access
b. Direct access
c. Sequential by secondary key
d. By record relationships
e. A combination of the above.

4. The subject of data dictionary/data directories and their place in the data base environment should be considered and fully understood by students.

5. The security provisions of database systems should be thoroughly covered. This should include coverage of the types of threats that exist to a database and the response or protection technique for the various threats.

6. The problems of concurrent processing and responses to the situations which can develop should be made clear to students.

7. Backup and recovery are most critical aspects of database processing. The various methods available within database systems should be thoroughly covered.

INSTRUCTIONAL DATA BASE SYSTEM

It is the firm contention and pedagogical philosophy of the author that the real understanding of students with much of the course content discussed above will depend on the ability to actually work with or utilize a database system in a case study or project environment. This is, of course, true of any computer concepts or programming class--the real knowledge and understanding is gained by doing, by "hands-on", if you will. All of this is based on the following general comments as they relate to a classroom situation. From a student standpoint, generally, the following is true:

1. I hear and I forget (lecture only)

2. I see and I remember (lectures supplemented by visuals and other aids)

3. I do and I understand (direct involvement, hands-on with a working system)

The Instructional Data Base System (IDBS) utilized is a subset of the CODASYL DBTG model and is intended to be used solely for instructional purposes. Performance is far too slow to compete with commercial database systems, and it was designed to process small databases.

IDBS Data Definition

IDBS does have most database capabilities. The smallest addressable unit of data is the field. Fields are grouped into records, and they in turn are grouped into files. A collection of database files is a database. Database records must have a unique primary key and may have any number of unique secondary keys which are maintained as inverted files. The IDBS database files can be related using the CODASYL DBTG set concept.

IDBS Data Manipulation

IDBS uses a subset of the CODASYL DBTG manipulation commands and can be accessed via query-language statements or subroutine calls in a batch operational mode. As an indication, IDBS has commands to OPEN and CLOSE the database created, CREATE and DESTROY files, sets and keys, FIND commands for record location. and PRINT, READ, STORE, MODIFY, DELETE, CONNECT and DISCONNECT commands to perform typical updating or modification activities against the database.

While its performance is slow and only small databases can be processed, IDBS does provide students with a real experience in working with databases. In a project environment, the student must define the logical structure of a database, create the database files and sets, load data into the database, and finally, perform typical updating of additions, deletions or modifications to the database.

The experience with this system has indicated studetns do indeed gain significant awareness of the important features and concepts of database processing by working with IDBS.

IDBS was developed by the author cited in reference 3 while at Colorado State University, and is available to other educational institutions at a nominal cost sufficient to cover reproduction and administrative costs. Further information can be obtained by communicating with the author.

SUMMARY

Database processing and the use of DataBase Systems in organizations is growing at an extremely rapid rate. It is important that educational institutions offer a course in this subject to meet the continuing advancing technology of the data processing industry. For a course to be meaningful, it should involve students in real or "hands-on" experience with a database system. Such involvement will illustrate and drive home the important concepts and knowledge which students should gain from such a course.

REFERENCES

1. Cardenas, Alfonso F., Data Base Management Systems, Boston: Allyn and Bacon, Inc., 1979.

2. Date, C. J., An Introduction to Database Systems, Reading, Mass.: Addison-Wesley Publishing Company, 1977.

3. Kroenke, David, Database Processing, Chicago: Science Research Associates, Inc., 1977.

4. Martin, James, Principles of Data-Base Management, Englewood Cliffs, N. J.: Prentice-Hall, Inc., 1976.

5. Ross, Ronald G., Data Base Systems, New York: AMACOM, 1978.

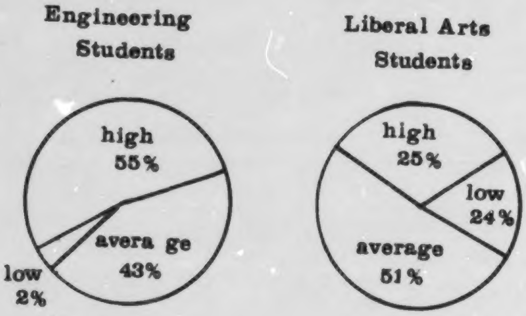
HUMANIZING THE INTRODUCTION TO COMPUTERS

L.V. Brillhart/A. Barnett
Triton College

INTRODUCTION

Triton College, a public community college serving the near west suburban Chicago area, has an "open door policy" for all entering students. As a result, any student declaring an engineering major is accepted into that curriculum and is allowed to take all courses which have no pre-requisites. Student background in aptitude and high school preparation covers the entire range from 7 to 33 ACT composite and from one year of algebra to calculus. As far as computer utilization is concerned, most students have never touched a terminal, although a few have extensive programming experience. The student's background requires that methods be used to humanize the gradual introduction of computers to engineering students in order to make the experience a valuable part of the learning process.

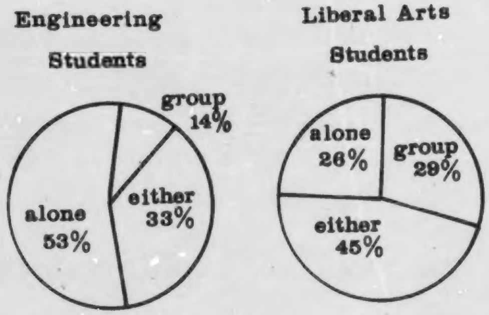
In computer programming, as in writing in any language, understanding audience and writing to that audience is of paramount importance. The programmer must consider the particular audience's attitudes, pre-requisite skills and experiences. Engineering students generally are a favorable audience who prefer content and learning modes compatible with computer utilization. For example, comparison of results on a Canfield Learning Styles Inventory[1] for the Numeric category between liberal arts students and engineering students are shown in Fig. (1).


Engineering Students

Liberal Arts Students

Fig. 1

The comparison clearly shows a greater preference among the engineering students for occupations requiring numerical ability. Furthermore, as shown in Fig. (2), 53% of the engineering students prefer to work alone (a condition appropriate to computer utilization), while only 26% of liberal arts students fall into this category.


Engineering Students

Liberal Arts Students

Fig. 2

The results of these studies[2] indicate that the use of the computer as an educational tool is very much compatible with the learning styles of engineering students. Any humanizing needed, then, is required to a greater extent for the benefit of the liberal arts students as well as for the benefit of those engineering students whose learning styles may substantially vary from the "typical" profile.

CURRICULUM

The engineering curriculum at Triton includes the following courses, as listed in Fig. (3), which incorporate computer utilization for some educational purpose.

Computer utilization in these courses is increasingly more complex, progressing at the same rate as the student's sophistication. For example, in rhetoric/engineering lecture, a team-taught course combining a one hour introduction to engineering with a three hour freshman composition course, all students must take only a computer generated and graded vocabulary test.

First Year

Semester 1	Semester 2
Rhetoric/Engineering Lecture *	Physics
Graphics*	

Second Year

Semester 1	Semester 2
Physics	Physics
Statics*	Dynamics*
Introduction to Automatic Digital Computers	

Fig. (3)

Starred (*) courses will be discussed in this paper.

Graphics, a semi-self-paced descriptive geometry and engineering drawing course uses computer testing to allow students to proficiency sections of the course whenever they are ready.[3] In addition, the computer is ulilized to check students' graphical solutions to descriptive geometry problems.

In the physics sequence, homework assignments are computer generated and graded. Interactive computer labs are also required of all students.[4]

Testing, similar to that used in graphics, is used to facilitate the semi-self-paced statics and dynamics courses. Additionally, students may elect to write computer programs relating to the subject area in partial fulfillment of the requirements of the statics and dynamics courses. It should be noted that students, while taking statics, are concurrently enrolled in "Introduction to Automatic Digital Computing," a course emphasizing FORTRAN with some work in BASIC, taught by the mathematics department.

Students receive a questionnaire at the end of each semester . In it they indicate problems with and attitudes toward the computer. At the end of each semester, changes are made in program in response to student's complaints. Results of three years of "humanizing" are discussed by topic rather than course.

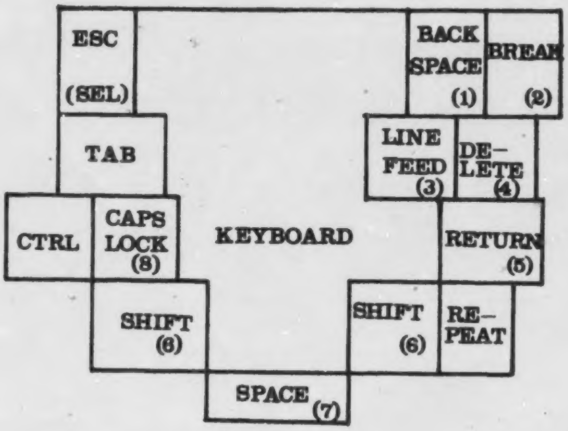
ACCESS

The computer system used by students at Triton is an off-line time-sharing system. Accessing is a rather complex multiple-step process In order to facilitate initial access to the system, each student receives a manual. This manual, written at Triton, consists of a step-by-step procedure written as a dialogue between student and computer, see Fig. (4).

1. Turn on terminal & Coupler
2. Dial Computer phone number and place phone in coupler cradle
3. Computer: TYPE CR
4. Student: Ɍ (Return key)
5. Computer: TRITON COLLEGE CANDE
ENTER USER CODE PLEASE
6. Student: COURSE/SECTION Ɍ
7. Computer: ENTER PASSWORD PLEASE
8. Student: DEPARTMENT Ɍ
9. Computer: ENTER CHARGECODE PLEASE
10. Student: Social security number (no spaces or dashes between digits)
11. Computer: BUDGET ACCRUED #SESSION TIME DATE
12. You are now in COMMAND MODE

Fig. 4.

The procedure shown in Fig. (4) is the result of a school-wide policy and reflects current security standards. The manual significantly helps first-time users respond easily to the computer's request for input. Since many of the students have never even seen a terminal keyboard, a partially annotated keyboard diagram is included in the manual. The function of the keys which are required or may be helpful is fully explained. A partial diagram with annotation is shown in Fig. (5).



1. BACK SPACE - the back space key may be used singly or repeatedly. You may then type on top of the previous wrong information. Only the latest character in a space will be transmitted.

2. BREAK - the break key stops the program. Do not break in the middle of a test. You will be credited as having taken the test once it is running.

3. LINE FEED - the line feed key advances the paper, one line at a time.

4. DELETE - the delete key deletes the last line entered.

5. RETURN - the return key is referred to as ℞. Hitting it transmits the last line typed to the computer.

6. SHIFT - the shift key causes the upper character on any key to be typed. Note, you do not need to capitalize anything, so do not use it to type capitals.

7. SPACE - the space key spaces the typewriter one space at a time. In the write up this is referred to as _.

8. CAP LOCK - the cap lock key should be pushed down.

Fig. 5

TEST PROCEDURE

Two types of procedures were developed for use in testing. The first procedure involves data input for obtaining tests. The second procedure incorporates a grading program.

1. Test. Once the students are in command mode, they follow a procedure which again is included in the manual, as shown in Fig. (6).

```
Student:  R MAKEUPTEST ℞
Computer: #Running 1111
#?
PLEASE ENTER YOUR STUDENT NUMBER

Student:  Student Number ℞

Computer:  PLEASE ENTER THE MODULE NUMBER

Student:  Module number ℞

Computer:  Runs test, GOOD LUCK

Student   BYE ℞

Computer  Prints costs
```

Fig. 6

All requests for data input are written in simple English. Should the student enter a wrong student number, the computer prints an error message, "Sorry, but I can't find your student number in that class." The student is then given two more opportunities to enter a correct student number. If all three attempts are wrong, the program will terminate. Before it shuts down, however, the computer will print the following message, "Why don't you take a break and find out your correct student number."

This type of procedure is followed by the computer for any student error. The program checks the input for validity. If the student has made a mistake in entering information, the computer prints an appropriate message identifying the invalid input and provides two more opportunities for the correct information to be entered. Before terminating because of three unsuccessful attempts by the student, the computer prints a final message suggesting a break. This procedure, with minor modifications, is used for all tests, see Fig. (7).

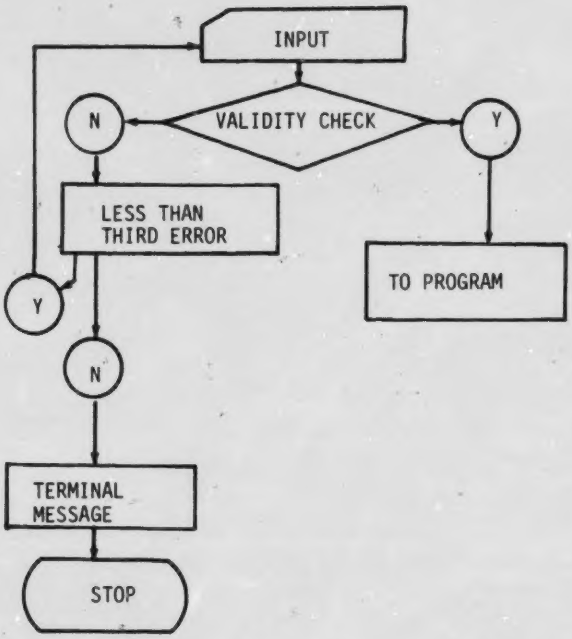


Fig. 7

All messages are instructive and are designed to alleviate confusion on the student's part. The appropriate message specifically identifies the invalid input. This basic procedure is used for all initial student input.

2. Answer. While testing for graphics and statics/dynamics is hand graded, the vocabulary tests used in the rhetoric/engineering lecture course are graded by the computer. The access procedure is identical to the test accessing procedure with the exception that the program called has a different name. The student then inputs his answers (multiple choice). The computer checks each answer for validity (within an A through D range). After all answers are entered, the computer prints the number of each incorrect answer. Students, therefore, are provided with immediate constructive feedback.

INTERACTIVE DATA PROCESSING

In graphics, as well as in statics/dynamics, some simple interactive programs exist. For instance, in graphics, students are asked to verify descriptive geometry solutions on the computer. The computer program provides comprehensive explanations for format of input data. Fig. (8) illustrates such a procedure used for a program which calculates the true length of a line.

```
THIS PROGRAM COMPUTES THE TRUE LENGTH
OF A LINE
GIVEN THE COORDINATES OF TWO POINTS
ON THE LINE IN THE V AND H VIEWS
GIVE  THE POINTS IN THE FOLLOWING
FORMAT:
ENTER A COORDINATE LIKE 2.3 AS 02.3
ENTER THE FIRST COORDINATE OF A_H
```

Fig. 8

The vocabulary used for the format explanation is identical to that used for description of points in graphics.

Since no files are manipulated, termination occurs only when the student enters a solution to be verified which is incorrect. The student is asked by the computer to repeat the graphical solution and to try again. If, at the second try, the student still does not enter a correct solution, the computer prints two messages:

1. Check your answer for the bearing in the program named BEARING and the slope in the SLOPE program.
2. See your instructor if the slope and bearing are also wrong.

The bearing and slope techniques are used in finding the true length of a line. Going back through the steps required in finding the solution helps a student determine the source of his or her error.

By the time students enter the statics and dynamics courses, they either have taken a FORTRAN/BASIC course (all engineering course programs are written in one of these languages) or they are concurrently enrolled. Less attention, then, is spent on format of input. Programs are still written, however, so that intermediate steps in the procedure are checked and messages indicating student errors are also written.

FILES

Up to this point, only errors of student origin have been discussed. At any time that files need to be manipulated, however, errors beyond student control can be encountered. Programmer mistake or machine malfunction can create these file errors. In addition, since all files are constantly updated by a number of people, chances of error increase. In normal operating procedure, errors of this nature would result in a program termination following a cryptic system error message. This is particularly frustrating to the novice who has no knowledge of system terminology. Even the student who has some knowledge of but no access to the files would have no recourse in this type of system termination.

In the programs used for the engineering courses, an attempt was made to replace the system messages written in language comprehensible to the student. The student is informed that he or she is not at fault and is directed to give the message to the instructor. After reading the message and consulting an instructor's manual (note: not all the instructors are computer experts), the instructor can locate and correct the file error without the services of a programmer or analyst. Errors beyond student control thus can be rapidly corrected without causing the student undue concern.

STUDENT PROGRAMS

Computer programs are written by some students to fulfill project requirements in statics/dynamics. The project must deal with subject areas in the engineering course in which the student is enrolled and is written for problems for which computer utilization is applicable. A specific instance would be a slider-crank mechanism velocity and acceleration program. A simple slider-crank mechanism is shown in Fig. (9).

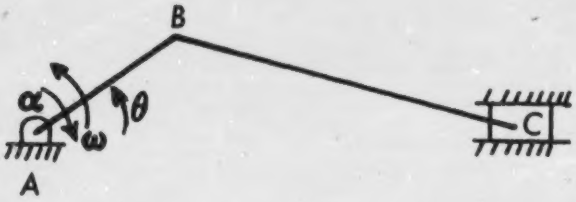


Fig. 9

A student has written a program in which velocities and accelerations of connecting rod (angular) and slider (piston, point C) can be found, if input data is given for crank length (AB), slider length (BC) and angular velocity (W) angular accelerations of crank. Other students running this program can either determine velocities and accelerations for specific crank angle with varying geometries or may obtain graphical output of system velocity and acceleration for crank angles at specified intervals.

Before a student's written computer project is accepted, documentation must be complete and understandable by a tyro programmer. Furthermore, possible student input data errors must be predicted and accomodated. For instance, if a crank length of zero is entered, a ridiculous answer should not be produced, but a message should be printed, such as, "Cranks cannot be 0 m. long, please re-enter the crank length." Error messages should also be included for alphabetic entries where numbers are required, or, for that matter, for omissions of entry. To be accepted, a student program may at no time terminate without a message, except obviously for reasons of equipment/phone line malfunctions. In this way, student programmers learn that their "customer" is a person with a finite tolerance for frustration.

CONCLUSIONS

While methods discussed in this paper have been applied to computer utilization by engineering students, they can, in fact should be applied to all programs. This is particularly important for students who have had no exposure to computers. The concept of dealing with a "black box", the computer is initially a frightening experience, therefore care must be taken to minimize negative feedback.

BIBLIOGRAPHY

1. Canfield Learning Styles Inventory, Humanics, Inc., Liberty Drawer 7970, Ann Arbor, Michigan, 48107
2. Brillhart, L.V. and Debs, M.B., "A Multi-Technique Approach for Effective Learning", Proceedings of the Conference on Improving University Teaching, 1979
3. Brillhart, L.V., "Modularized Self-Paced Graphics at the Community College", Proceedings of ASEE Conference, 1979
4. Brillhart, L.V. and Shawhan, D., "Computers: New Avenues for Engineering Students at the Community College", Proceedings of NECC Conference 1979

"STANDARD" REVIEW UNITS IN A CAI MATERIALS APPRAISAL PROCESS

Jeanne Burson, Ph.D.
The Ohio State University
College of Medicine
Division of Computing Services for
Medical Education and Research

ABSTRACT

Results of a study testing a CAI program rating mechanism, as part of a materials appraisal process, are presented. Reviewers rated various programs on a scale of one-to-ten on each of seven criteria: appropriateness, responsiveness, cohesiveness, correctness, completeness, usefulness and the learning experience. Variability of the ratings within and between program reviews were determined. The highest acceptable variation was ± 2.0 standard deviation, regardless of type of program, status of reviewer or individual application of the criteria. The ratings combined with reviewer comments and a program description should provide composite information on which CAI users may readily select specific programs appropriate for them.

INTRODUCTION

The Computer Assisted Instruction (CAI) system at The Ohio State University College of Medicine has grown from a one-terminal, one-course system in 1967 to the operational system of today, which:

- provides 22 hour per day, seven day per week CAI access for students and health professionals throughout the United States;
- maintains a program library of over 450 interactive hours of CAI materials;
- logs approximately 2,400 usage hours/month; and
- adds CAI program materials to its library at the average rate of 2.25 interactive hours per month.

It is meaningful to note that the emphasis in CAI material has been placed on: a) the quality of content; b) the applicability and usefulness of the materials in the health sciences curricula and c) the soundness of the educational strategy employed. The approach has been not to emphasize CAI as a technology, but rather to develop CAI as an educational resource which can uniquely meet some needs. With these parameters specified, a method for appraisal of the CAI materials was indicated.

BACKGROUND

A CAI Materials Appraisal Project (MAP)* was conducted as a 28-month pilot study to: 1) design and develop a model for the appraisal of CAI materials; 2) implement the appraisal process by submitting selected CAI programs to the process; 3) develop and implement mechanisms for publishing and/or disseminating the informative reviews; and 4) evaluate and modify the appraisal process to increase its operational effectiveness and efficiency.

Personnel support for this project was provided by the staff of the Division of Computing Services for Medical Education and Research (DCSMER), the Division of Research and Evaluation in Medical Education (DREME) and the Independent Study Program (ISP) of the College of Medicine.

Original project objectives emphasized the development of a process for evaluating CAI materials. The design and development of the assessment device was itself an instrumental activity necessary for pilot implementation of the total evaluative process. The actual development of an instrument was an endeavor that, although undertaken on a task basis by the Division of Research and Evaluation in Medical Education, reflects decisions made by the entire project staff. The resultant instrument has been greatly influenced by decisions relative to process considerations. The project staff was involved in two very pragmatic and crucial activities regarding instrument development: (1) the identification of data points necessary, and, hopefully, sufficient for evaluative purposes, and (2) the subsequent design of potential means (both items and format) for accessing information relative to the data points.

The final step in the development of the appraisal instrument was an attempt to validate the Materials Appraisal Forms by submitting these to the original pool of consultants. The consultants were requested to indicate a) whether the items adequately addressed the specified criteria, b) whether the items satisfactorily evaluated CAI materials, and c) whether any items should be added or deleted. Consultants' responses were analyzed and synthesized to produce the final forms of the pilot instrument used in the CAI Materials Appraisal Project. This instrument can be said to demonstrate face validity.

---

* Study supported by National Fund for Medical Education (Grant #81/75), 1976-78. Principle investigators A.J. Merola, Ph.D. and Ruann E. Pengov, Ph.D.; Manager, Jeanne L. Burson, Ph.D.

Individual reviewers pinpointed several weaknesses characteristic of the instrument. Most of these can be classified as editing/revising problems. Some questions were redundant and, therefore, easily eliminated. These questions were eliminated by careful editing and revision of certain items.

Another type of problem was not as easily remedied. Certain items required subjective opinion; more explicitness in how to assess the presence or absence of an attribute must be provided the reviewer if such subjectivity is to be avoided. The use of descriptors such as "reasonable", "sufficient", "adequate", or "appropriate" creates a situation in which each reviewer must individually define the minimum acceptable standard. It was questionable whether resultant data is truly comparable across reviewers. Certain reviewers alluded to this problem within the context of absolutes.

It was determined that there would be four (4) areas for review: content, strategy, technical and participant. A separate review was made for simulations. Reviewers were identified for each of seven programs selected in accordance with stated reviewer qualifications. Local reviews were utilized for the tryout and Phase 1 of the review; persons external to The Ohio State University (primarily Health Education Network members) were selected for Phase 2. A fee schedule based on the average interaction time for each program was formulated and the appropriate fee paid each reviewer. Access time for reviews was also paid by the project. The total process involved 51 reviewers in eight institutions. Utilizing local reviews for the tryout enabled the staff to interface with the reviewer during and following the actual review. This interface proved valuable in revising the process and finalizing the informational sheets accompanying the forms. However, it was evident, even at this stage of the project, that personal interface, at least by telephone, would be required to facilitate access to the program and return of the completed critique. Therefore, monitors identified at each external institution, were contacted and clarification of the project and program access procedure was given.

MODEL

A viable review process model (Figure 1) evolved resulting in a prospective rating form with review comments and summary analyses.

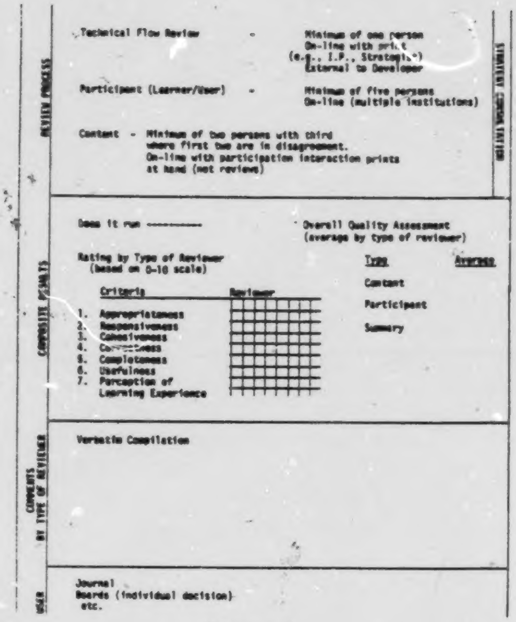
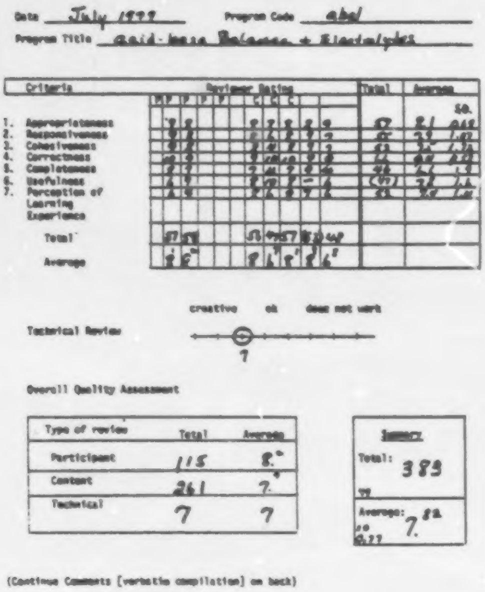

REVIEW PROCESS
Technical Flow Review - Minimum of one person
On-line with print
(e.g., I.P., Strategist)
External to Developer
Participant (Learner/User) - Minimum of five persons
On-line (multiple institutions)
Content - Minimum of two persons with third where first two are in disagreement.
On-line with participation interaction prints at hand (not reviews)
STRATEGY CONSULTATION
COMPOSITE RESULTS
Does it run ----------
Overall Quality Assessment
(average by type of reviewer)
Rating by Type of Reviewer
(based on 0-10 scale)
Criteria
Reviewer
1. Appropriateness
2. Responsiveness
3. Cohesiveness
4. Correctness
5. Completeness
6. Usefulness
7. Perception of Learning Experience
Type
Average
Content
Participant
Summary
COMMENTS BY TYPE OF REVIEWER
Verbatim Compilation
USER
Journal
Boards (individual decision)
etc.

Figure 1

CAI MATERIALS APPRAISAL PROCESS (MAP) MODEL

The first step is the technical review -- does the program run or not? This review is done by one person online with a print at hand. After that, it is reviewed by participants, target or collateral, with a minimum of two of those persons reviewing it online. Next, the content reviewers review it online, utilizing the participation interactions to verify the content as correct. There should be a minimum of two content persons reviewing and a third if the first two were in major disagreement. At this point, a CAI instructional strategy consultation might be indicated or desired for assessment of strategies involved.

A total of at least five reviewers (or more if needed) rate the program on a scale of one to ten on each of the following items: appropriateness, responsiveness, cohesiveness, correctness, completeness, usefulness and their perception of the learning process. This scale is then averaged and recorded for each criteria and summarized for each reviewer. The scaled result is included with the open-ended comments of the reviewer and a summary analysis by type of reviewer as to the overall quality of the program. The composite of information should allow the users to make the decision as to whether or not the CAI program is appropriate for them. Figure 2 is an example form.



Date July 1979 Program Code abel

Program Title Acid-base Balance + Electrolytes

Criteria

1. Appropriateness
2. Responsiveness
3. Cohesiveness
4. Correctness
5. Completeness
6. Usefulness
7. Perception of Learning Experience

Total

Average

Technical Review: creative — ok — does not work; 7

Overall Quality Assessment

Type of review	Total	Average
Participant	115	8.
Content	261	7.
Technical	7	7

Summary

Total: 383

Average: 7.82

(Continue Comments [verbatim compilation] on back)

Figure 2

CAI MATERIALS APPRAISAL PROCESS ANALYSIS

PILOT STUDY: SUMMARY COMMENTS

As anticipated, it was very difficult to secure reviewers. CAI is not yet looked upon as being a "legitimate scholarly endeavor". However, offering a fee for services did facilitate the process. Cost: utilizing a review fee based on average program interaction hours, paying monitors, and adding contact fees, the cost for the seven programs (in the first study) with 51 reviews, averaged $70.00 per review (average $510.00 per program reviewed). Using shorter programs decreased the cos in the second study.

In implementing the process, little difference, aside from communication costs (online and personal), could be seen whether utilizing "in-house or external reviews". In-house reviewers, as a group, were very critical and showed little bias. With the exception of system access, only a few problems were encountered with external reviewers.

The review procedure succeeded in identifying program strengths and weaknesses. The most helpful information, the isolation and description of specific strengths and weaknesses, was derived from evaluators' "comments" and the "narrative critiques". Because the question response scale was essentially a two-level scale ("agree" or "disagree"), the evaluators' question rating supplied a very limited evaluation but served to "pin-point" problems and/or strengths across reviewers.

Writing reviews from the form responses was possible; however, the procedure was time-consuming. Reference to the comments is mandatory if a review writer is to include substance in the review. The ability to quantify responses on a scale enabled the design of a more substantial and concrete review (see Appendix A for an example Sample: Analysis and Comments).

MATERIALS APPRAISAL PROCESS RATING ANALYSIS (MAPRA)

In March, 1979, a one-year continuation study was undertaken by the Division of Computing Services for Medical Education and Research, utilizing the material appraisal process, to field test the review rating mechanism.

During the first five months of the project, reviews were elicited from 22 reviewers in three health care institutions. Three programs, two tutorial presentations and one simulation, were involved in this first phase of the review. Efforts were made to: 1) determine the relationship of the composite ratings to the individual item responses, and 2) determine the variability/deviation of the responses within and between program reviews.

Table 1 contains the mean score and standard deviation for each of the three programs. The results given are for reviews within each program and between the programs on each of the seven criteria. It should be noted that the largest variation occurred in responsiveness (1.35) and correctness (1.49) and the smallest in completeness (0.21) and usefulness (0.26). However, in this first phase of the project, the highest variation overall did not exceed 2.0 standard deviations, perceived to be an acceptable deviation from a mean rating, regardless of type of program (tutorial or simulation) reviewed or status of participant (target or collateral) making the review.

Table 1

MAP RATING ANALYSIS
Mean-Standard Deviation (SD)

(Phase 1 Results)

	Within Programs						Deviation Between Programs (overall)	
	Mean			SD				
Criteria	case	abel	rescue	case	abel	rescue	Mean SD	SD
1. appropriateness	8.7	8.1	8.3	1.2	0.69	1.8	1.23	0.55
2. responsiveness	6.8	7.9	8.0	3.6	1.07	1.5	2.08	1.35
3. cohesiveness	7.0	7.5	7.3	2.96	1.72	1.2	1.96	0:9
4. correctness	7.7	9.4	6.1	3.39	0.53	2.7	2.21	1.49
5. completeness	8.5	8.6	7.6	1.64	1.9	2.07	1.82	0.21
6. usefulness	8.8	7.8	7.7	1.6	1.6	2.08	1.75	0.26
7. perception of learning exp.	8.5	7.4.	7.8	1.08	1.4	1.72	1.39	0.34

During the second phase, additional reviewers were utilized to compare ratings with and without student printouts for online content review. The final review phase of the project focused on the process itself, as well as accumulating additional data to assess the rating concept.

During Phase II, three additional statistics were applied. The first, the standard error of the mean (SEM) was used to calculate the deviation of sample mean caused by errors of measurement and by sampling error. The second, a test for differences among several independent variances (F-maximum Test for Homogeneity of Variances, F_{max}) was used to determine if the variances of each program and each criteria were homogeneous. The third, a test for differences between variances of two independent samples (F) was used to determine if there was a significant difference between variances of the content and participant reviewers in each program. The preliminary* results are presented in Tables II and III.

Table II

PROGRAM VARIANCES AND F_{max} (By Criteria and Overall)

Variances by Criteria

Program	1	2	3	4	5	6	7	Overall
'abel'	0.48	1.14	2.95	0.28	3.61	2.56	1.96	1.93
'rescue'	3.24	2.25	1.44	7.29	4.28	4.24	2.95	3.53
'fammed'	2.67	2.00	7.50	5.30	5.17	2.50	7.67	3.49
'solve'	1.50	4.83	7.80	3.30	3.50	13.50	11.50	3.49
'case M'	3.67	2.67	2.83	2.00	4.72	6.16	4.80	4.40

- - - - - - - - - - - -

F_{max} (Between Programs)

	1	2	3	4	5	6	7	Overall
	7.60	4.24	5.42	26.00*	1.48	5.40	5.87	2.27

*df = 6; number of variances 5; significant at .01 level

There were no significant differences in the overall variances of the program, but when applied to the individual criteria across programs, the F_{max} of 26 for the correctness criteria was found to be significant at the .01 level, indicating that the variance was indeed not homogeneous.

When considering the correctness criterion, there was found to be different variances by the content reviewers (at .01) between the various programs; and the variances were also significantly different between the content and participant reviewers in the 'case', 'solve' and 'rescue' programs. The latter was significant only at the .1, .1 and .2 levels respectively. The participant reviewers were very consistent with a range of variances of 5.67 in the completeness criterion to 1 or 0 in many of the other criteria.

Table III

F_{max} or F BETWEEN STATUS OF REVIEWER

F_{max} Between Status of Reviewers (Between All Programs)

Content Reviewer	Criteria 1	2	3	4	5	6	7
Max - S^2	5.50	5.30	11.00	8.50	7.25	10.50	11.00
Min - S^2	0.50	1.25	2.25	0.25	4.25	2.67	1.75
F_{max} (6/5)	11.00	4.24	4.89	34.00^a	1.70	3.90	6.29

a = F_{max} significant @ 0.01 level

Participant Reviewer	1	2	3	4	5	6	7
Max - S^2	3.70	1.00	4.00	1.00	5.67	5.00	4.30
Min - S^2	0	0	1	0	0	0	0
F_{max}	NOT CALCULATED						

- - - - - - - - - - - -

F Between Status of Reviewer Specific to Program (Criterion 4 Only)

	F for #4
'abel' (df = 5/2)	0.00
'rescue' (5/4)	2.80^c
'fammed' (5/2)	2.43
'solve' (4/3)	8.60^b
'case' (n) (3/4)	4.50^b

b = significant @ 0.1; c = significant @ 0.2

SUMMARY

Studies were undertaken to develop a CAI Materials Appraisal Process that would culminate in composite information which CAI users might readily utilize in selecting specific programs appropriate for them. A validated appraisal form, designed to elicit information for dissemination and possible materials revision, was field tested.

From the preliminary data, it was perceived that in spite of the increased variances in content reviewers, an acceptable overall mean rating on seven appraisal criteria could be obtained regardless of the status of reviewer, type of program or individual definition of the criteria. However, between type of program and between the status of reviewers, the variances in individual criteria might not be homogeneous. Additional data to be available at the completion of the present study should confirm or reject this hypothesis. The isolation and description of specific program strengths and weaknesses, made possible by the design of the form, proved to be valuable tools for writing reviews for dissemination and for use in materials revision.

AFTERWORD

The reviews of each program submitted to the process will be summarized. A judgmental analysis of the usefulness of the rating process will be elicited. The acceptance by publishers, editorial boards, educators and potential users of a review process, which includes a mechanism for providing assurance that the materials have been judged on specified standards for quality, is essential. This acceptance can potentially enhance the quality of computer-based education materials. Peer appraisal and review standardization has the potential to bring about change in a number of key elements such as increasing faculty (author) incentives through documentation of creative scholarly endeavors. However, the ultimate evaluation of the studies will be the actual acceptance of the results by potential users of the CAI materials -- and this result can only be determined through the passage of time.

---

* all reviews have not been returned. Final conclusions pending completion of the project.

Appendix A

EXAMPLE: ANALYSIS AND COMMENTS

PROGRAM CONTENT

The organization of Module Q content was rated by the vast majority of both educators and participants as good. A participant remarked that generally the program was excellent and that the only problem was the lack, perhaps, of "short cases" of clinical material to reinforce the clinical aspects of pathology. Educators evaluated content sequencing as "good"; participators, less unified in opinion, spread ratings from among "excellent", "good", and "adequate". Two participants suggested the program lacked cohesiveness; specifically, they reported that the four sections were not connected and that progression through submodules was not in a design enabling students to think through problems. The majority of participants did state content was logically presented. Educators and participants agreed that content was accurate; educators further stated that content was current. Ambiguity in some program content was cited by several participants. These evaluators said, that in some instances, the intent of some questions was unclear and that confusing answers existed. A few participants and educators also felt gaps in program content existed (especially in submodule A). The majority of reviewers did not cite program information gaps. With regard to learning activities, one educator indicated that in some areas within submodules information was not covered. Other educators did not note deficiencies in learning activity information. All participators reported that examples providing practical applications of knowledge were given. A need for more examples and more complex examples was indicated. Educators also identified a need for providing additional references. References which were included were, for the most part, evaluated as appropriate.

PROGRAM INSTRUCTIONAL DESIGN

Strategists and participants reported learning activities were appropriate; they indicated alternative instructional strategies should be considered. Strategists did not feel the program included unique learning or performance activities. Most participants indicated 'module Q' included a variety of learning or performance activities. One participant said the program demonstrated good variability in format and conceptual design. Most educators and participants also stated that interaction between the participant and program was continous. A negative reaction towards the interaction was logged by a participant who said program reiteration on correct answers was excessive. Strategists noted 'module Q' acknowledged participant achievement. They also indicated the program enhances self-evaluation but does not maximize individualization. 'module A' was identified as a very rigid tutorial evaluation by one strategist.

Feedback, according to participants, was responsive to input, was pertinent and did anticipate various input. Comments summarized feedback as generally good and helpful. Room for some improvement was noted, however. Strategists reported feedback did anticipate participant response reasonably well. Participants recorded 'module Q' as personalized to the participant. In some instances, feedback was described as forced or unnatural in syntax.

The type of questions used were viewed by both strategists and participants as whether varying in a beneficial manner or varying so as not to detract or add to the program. Participants liked the variability of questions but indicated that fill-in-the-blank questions took too much time to complete. With regard to the presentation of questions, strategists indicated questions varied in a beneficial manner.

Strategists indicated that participant performance data was not maintained. Due to insufficient information, strategists were unable to offer a cost analysis comparing this CAI program with other educational approaches aimed at reaching the identified objectives.

PROGRAM TECHNICAL DESIGN

All participants indicated the program was conducive to individual learning. Qualifying comments characterized the program's feedback as very helpful for review and the program's capacity for individuality as limited by predetermined learning priorities. Strategists did not feel optimum use of unique CAI system capabilities was employed. Participants did not evaluate 'module Q' as allowing varying opinion. Technical design reviewers and the vast majority of participants indicated the program allowed for reasonable typing errors and synonym use. Comments cited 'module Q' as excellent and flexible in this regard. Technical reviewers reported the program responded to input within a reasonable length of time; strategists cited programming as effective. With reference to programming errors, one closed loop was found. Strategists indicated 'module Q' could be easily modified for addition or deletion of material. Technical reviewers did not note superfluous or inefficient commands.

Coding, as judged by technical reviewers, was seen as correct within system and language limitations, allowing for variation in participant response, appropriate for the program's intent, allowing for presentation variation, well documented, and maximizing individualization. 'module Q's internal organization was judged as permitting instructor control and management and allowing for ease of modification and maintenance.

Technical reviewers also saw the program as utilizing the unique capabilities of the computer system and language. System features such as help, restart, and review were noted. Technical reviewers also reported the program did not require a special terminal.

PROGRAM EDUCATIONAL MERIT

Participants reported that program entrance level and prerequisistes were stated; the majority of educators disagreed. Most members of both groups indicated the intended participant was specified. Educators and participants did agree that 'module Q' was suitable for the intended audience. Prerequisites were judged as appropriate by most educators. One member of the group felt prerequisites might be inappropriate.

Both review groups indicated that learning objectives were clearly identified; educators also said objectives were adequate in depth. Participants cited program length as suitable for meeting the stated objectives; educators disagreed.

Learning activities were judged by educators as adequate and presented in an optimal sequence for learning. Most participants also stated that most of the learning activities were needed for attaining the stated objectives. Educators and participants agreed that sufficient opportunities for reinforcement were available; educators reported the opportunities as adequate. Most educators also felt that opportunities for assistance within the program were adequate.

Feedback to participant response was judged by educators as responsive and pertinent. An educator commented that feedback was somewhat mechanical and redundant but better than feedback of some other programs. Educators also indicated that 'module Q' stimulated thinking; all judged the program as more stimulating to learning than a classroom experience. Most educators also determined that 'module Q' motivated study.

In evaluating the quality of program information, educators offered a variety of opinions. In judging quantity, educators said 'module Q' provides more, the same amount, and less information that a classroom experience. An educator commented that learning is more active than in a classroom. The majority of educators also believed the program was sufficiently documented to allow an instructor to prescribe all or parts of the program for individuals.

PROGRAM'S ADDITIONAL LEARNING RESOURCES

Participant evaluators did not rate the quality of visuals as high. Comments indicated that some of the diagram arrows were confusing. Strategists and participants did not judge visuals as essential to learning. A participant remarked that the visuals were helpful, but not always essential in many areas of study. Participants were not consistent in judging if the visuals enhanced reinforcement; opinions were both pro and con.

AN ON-LINE INSTRUCTOR CONTROLLED MANAGEMENT & REPORTING SYSTEM WITH SELECTABLE "TRACKS"

Judith Contolini
The Ohio State University
College of Medicine
Division of Computing Services for
Medical Education and Research

"The Physiology Self-Evaluation Units" ('physeu') is a computer-assisted instruction (CAI) program used in the first year of The Ohio State University Medical School's Physiology curriculum as a means of self-evaluation and self-instruction.

Physiology was originally presented in the traditional lecture format. A disadvantage of this traditional method was the student's passive behavior. The student seriously studies Physiology only before examinations. In many cases, it was only then that the student realized his deficiencies. The 'physeu' program was developed in order to involve the student earlier and to pace his study efforts more evenly, and thus, enhance learning.

In the present format, the students attend lectures and read their textbook and faculty developed handouts on a given content area. Once this is done, a 'physeu' unit is released to them. Each unit is to help the student review material already covered or previously learned. The encounter, therefore, will be one of stimulus (question), interaction, and reinforcement of related information.

The 'physeu' program currently includes 15 content units and four review units. Each content unit contains an A and B section, similar in content, with approximately 25 questions in each. The review units, with approximately 25 questions each and pertaining to three or four selected content units, are used to assist the student in reviewing and integrating the materials from these selected units.

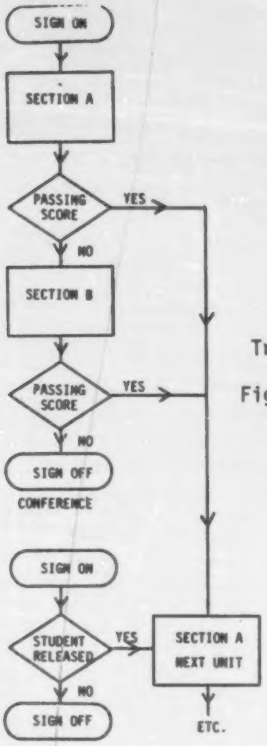
Some basic requirements went into the design of 'physeu'. First, students should be able to move at their own pace, independent of other students. Secondly, instructors would need to assess student progress on-line, without interruption to the students. Student flow through the program should be flexible; instructors should be able to tailor the presentation of 'physeu' units for their individual classes. Finally, some students who fail to perform satisfactorily should be counseled by their instructors prior to continuing progress through the content units.

These requirements are met by the 'physeu' management and reporting system. An individual file is maintained on each student. The file allows the student to sign off anywhere in the program and take-up again where he left off. It stores information which can be used by the instructor to generate a variety of reports on either an individual student or the class as a whole.

Each student's flow through 'physeu' is determined by the instructor's selected sequence of 'physeu' units and the selected teaching mode, or "track". Because the topics in physiology are fairly fixed, covering areas such as cardiovascular, respiratory, etc., the order in which these topics are presented might be quite different depending on the instructor or the individual school's curriculum. The instructor's selected tracking mode for his class will determine if 'physeu' will be used as an integral part of the curriculum or simply as an adjunctive learning resource. At present, there are three basic tracks provided. The tracks vary from strictly controlled sequencing established by the instructor to free access of any content unit.

Tracks 1 and 2 are strictly controlled-sequential-access tracks (Figures 1 and 2). The instructor selects the exact unit sequence for his students and may also determine when each unit is to be available. Every student begins with Section A of the unit. If the student attains a satisfactory score, he is free to go on to Section A of the next unit in the sequence. However, if the student does not pass Section A of the unit, he is branched to Section B of that same unit, which covers similar material, implying that more study and/or review is necessary.

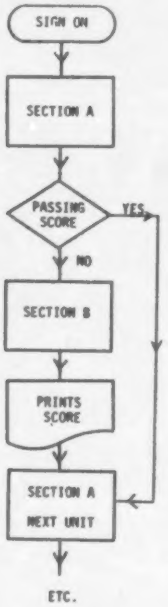
An additional Track 1 feature places a student who has failed both Section A and B of a unit in "lock-out" status. This prevents a student from progressing further in the sequence until he has had a conference with the instructor to clarify areas of difficulty. Following the conference, the student is "un-locked" on-line by the instructor. The student may then continue with the next unit of the sequence.


SIGN ON
SECTION A
PASSING SCORE
YES
NO
SECTION B
PASSING SCORE
YES
NO
SIGN OFF
CONFERENCE
SIGN ON
STUDENT RELEASED
YES
NO
SIGN OFF
SECTION A NEXT UNIT
ETC.

Track 1

Figure 1

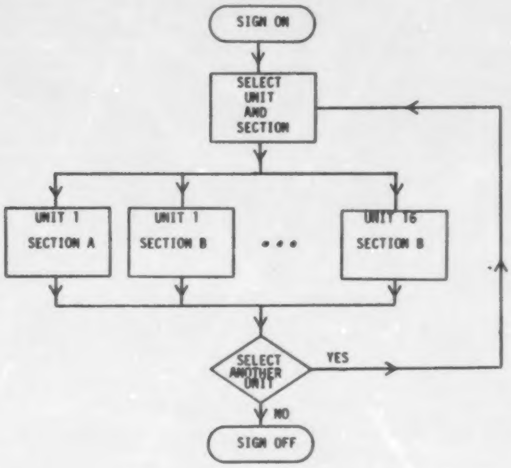
If the student is in Track 2 and fails to pass Section B, the student will just continue with the next assigned unit. The "lock-out" feature is not employed.


SIGN ON
SECTION A
PASSING SCORE
YES
NO
SECTION B
PRINTS SCORE
SECTION A NEXT UNIT
ETC.

Track 2

Figure 2

Track 3 is a select mode (Figure 3). The student may select any 'physeu' unit and sections A, B or both if desired. The achieved level of performance has no effect on the student's progress. As in Tracks 1 and 2, essential information about the student's status and unit performance is stored.


SIGN ON
SELECT UNIT AND SECTION
UNIT 1 SECTION A
UNIT 1 SECTION B
...
UNIT 16 SECTION B
SELECT ANOTHER UNIT
YES
NO
SIGN OFF

Track 3

Figure 3

Besides the three "tracks" that provide student management, a 4th Track, review mode, is also available. This track permits free access to any unit or section desired. However, no records are maintained, no scoring is done, and reports on student performance are not available.

Two additional features can also be incorporated into Track 1 and 2 as part of the instructor's sequencing, a Review Unit and a Review Point. At the conclusion of three or four units that encompass a related subject, for example cardiovascular or respiration, the student may be given the opportunity to review more material on the subject, a Review Unit. The questions in these units are similar but different from those presented in the related units. In a Review Unit, beginning and ending dates are recorded, but there is no scoring. The student's location in the unit is also saved if he wishes to sign off before completing it.

If the instructor wishes to allow the students to review previous material before completion of all the units of a related subject, he may designate a Review Point where a student may request to review completed units. The Review Point allows students to return to as many previous units, Sections A, B or both if desired, thus permitting the student more review and drill of specific areas.

If a student takes advantage of a Review Point, no previous dates or scores are affected. At the completion of every unit reviewed, the student returns to the Review Point where he may choose another unit to review or continue within the 'physeu' sequence.

The individual student files are accessed via an I.D. number assigned by the instructor at time of student registration. Once this is done and the instructor has entered his selected unit sequence into the computer, the students are ready to begin. The on-line reporting and management system can now be used. When student records are initiated, the instructor may begin to access the information regarding the students' status on the 'physeu' program. The instructor is provided with a set of commands to maintain and print the student's progress and performance. These capabilities are achieved on-line through the use of instructor commands, accessible only by the instructor's number and a password. The 'physeu' commands permit the instructor to fashion the 'physeu' program so it can serve a number of instructional needs without resorting to unique programming of specific requirements for every instructor or curriculum.

To maintain student records the instructor, or some agent of the instructor such as a teaching assistant or secretary, has the ability to correct misspelled names (alter command) and change incorrect student scores (change command). Student records can also be removed if students drop out of the curriculum (delete command). Through the use of these commands, the instructor selects the desired track (track command), inputs his unit sequence for Track 1 or 2 (sequence command), releases students who have been "locked-out" (Track 1) (release command), and holds units until their time of availability (hold command). The instructor can also request a list and description of all units (list command). Because of the wide variety of commands, each command is explained on-line by simply typing "/help" after the desired command has been requested.

The progress and status of students are obtained by using the report command. There are five available reports: roster, unit, student, progress and conference.

Roster Report: This report prints a list of the students registered, their associated group code and their four-digit I.D. number. This report may be used after the registration command to get a list of students. Since the students must know their four-digit I.D. code to sign-on to 'physeu', the roster report may be used as a source from which the I.D. codes are distributed. Additional roster reports may be produced for other instructors associated with the class.

Unit Report: The unit report is designed to provide the instructor with scores on any selected unit for all students in the class or for a group of students. Both Sections A and B scores are reported.

Student Report: The purpose of this report is to print an individual student's scores on one or more units, Sections A and B.

Progress Report: This report lists the students' progress on any given unit and section desired. The names and I.D. numbers are listed for those students who have not attempted the unit and section.

Conference Report (Track 1 only): This report is designed to list those students who have not obtained a passing score on the B section of a unit. This is a cue for the instructor that the student is experiencing problems and perhaps additional requisites are necessary. The students on this list will not be able to continue with the program until the instructor releases them by using the release command.

The present design of 'physeu' is very successful for the medical students at The Ohio State University in the Track 1 mode. As a result of outside usage and needs of other curriculums, additional features to the program have been requested. For those curriculums wishing to use Track 1 or 2, a command could be provided to allow the instructor to include only Section A or only Section B of a unit. The instructor may also wish to let the student choose the section. If the student does not pass the one section he will be "locked-out" (Track 1) or will just continue with the next unit (Track 2).

Track 3 can be made more flexible if a student has the option of choosing a new unit each time he signs on to the program. At present, if he signs off in the middle of a unit, he returns to that exact location and cannot choose another unit until its completion.

A new command can be created for the purpose of summarizing all students' performances. The list would be alphabetical stating the completed units and scores for each student.

In the future, Review Units will probably be expanded and generalized. Questions related to the content units will be grouped according to subject area and stored in a data bank. The instructor may then include a Review Unit after any set of units. A Review Unit will be compiled based on the instructor's selection of subject areas and the number of questions selected for each subject. The questions would then be randomly selected from each area creating a unique unit for each student.

'physeu' has been accepted by the medical students as an excellent adjunct to learning. In the questionnaire distributed each year at the completion of the Physiology program, more than 90% of the students have said that 'physeu' has enhanced their learning of physiology and helped to clarify concepts and material presented in lectures. At least 90% also felt 'physeu' was helpful in forcing students to study on a continuing basis and keep a pace with material presentation.

This overall approach to the teaching of Physiology appears to have made a significant difference in the medical students' performance of Physiology over previous years. This is reflected in the students' achievement in the National Board Examination Performance (shelf copy). Using the mean score of the 1970 Physiology class at The Ohio State University, the mean scores from the 1971-1975 classes (before 'physeu') increased by ten (10) standard score points. However, the mean scores from the 1976-1978 classes (with 'physeu') increased by 50 standard score points.[1]

[1] Lipsky, Joseph A., et al. "Physeu: A Computerized Instructional and Self-Evaluation Program for Learning Medical Physiology." *The Physiology Teacher*, 22:5, pp. 31-35, October, 1979.

REFERENCES

Dennis, Verl E. and Stephen E. Hewitt. *Operating Manual: Physiology Self-Evaluation Units ('physeu') Management and Reporting System*, 1979.

Division of Computing Services for Medical Education and Research, The Ohio State University, College of Medicine. *Interim Progress Report #2 -- The Phase II Physiology Self-Evaluation Units 'physeu' Program*, 1978.

Lipsky, Joseph A., Lawrence L. Gabel, John J. Curry, Margaret T. Nishikawara, and Heinz P. Pieper. "Physeu: A Computerized Instructional and Self-Evaluation Program for Learning Medical Physiology." *The Physiology Teacher*, 22:5, pp. 31-35, October, 1979.

MICROS-BY-THE-DOZEN

B. D. Cross
Dr. Joseph E. Barbay
Dr. Dale H. Besterfield

A LOOK AT SIU

Southern Illinois University at Carbondale is a major university founded in 1874 and currently enrolling over 22,000 students. Bachelor, masters, and doctorate degrees are offered in over 90 different areas including law, medicine, engineering and technology, business, agriculture, science, fine arts, etc. The Industrial Technology Program was initiated in 1943 and is accredited by the National Association of Industrial Technology. Industrial Technology Military Programs commenced at Travis Air Force Base, California in 1975 and is now offered at 12 military installations.

INDUSTRIAL TECHNOLOGY PROGRAM

The program delivered to military bases is essentially the same as that offered on campus and is staffed by full-time, on-site university professors and by campus faculty who are flown in for weekend teaching. On-site faculty personnel function both as instructors and program advisors for the students. Classes are delivered to students in a weekend format so active duty military personnel and authorized civilian employees can participate with minimum conflicts on normal work schedules. Some courses may require an occasional meeting other than the normal weekend schedule. A typical three hour course would meet every other weekend for a total of three weekends. This can be compared to a normal semester of 12 semester hours credit over 16 weeks.

The Industrial Technology program is an excellent vehicle for obtaining competencies useful for in-service and post-service career development. Both during service and as a second career, industrial technology positions are associated with production planning and scheduling, process design, quality control, methods analysis, personnel supervision, material and equipment procurement, facility planning, equipment design, job estimation, technical sales, maintenance supervision, safety, and other production related functions.

PROGRAM FEATURES

The Industrial Technology degree offered by Southern Illinois University at Carbondale has the following features.

- Industrial technology courses (48 semester hours) which can be completed in 16 months (4 semesters).
- General Education courses which can be completed with any institution at any base or by proficiency.
- Credit for military schooling.
- Credit for work experience.
- Credit for proficiency through CLEP (general examination) and DANTES.
- Full Veteran Administration reimbursement.
- A capstone option (qualified students have reduced general education requirements).
- Students with a baccalaureate oriented associated degree from an accredited institution will have general education requirements completed.
- Students enjoy the same rights and priviledges as full time on-campus students including the university placement service.

DEGREE REQUIREMENTS

Core Requirements

These are the Industrial Technology courses offered by SIU-C and includes motion and time study, materials handling and plant layout, production and inventory control, quality control, industrial safety. This program is a balanced curriculum of studies drawn from a variety of disciplines relating to industry to prepare men and women for managerial responsibilities in government, industry, and business. Thirty-one semester hours are required.

Technical Electives

These courses include industrial internship, occupational safety and health standards, special problems in industrial and technology, technology design and other elective courses. This area also includes credit awarded for work experience, military schooling, and other technical classes transferable to SIU-C. Forty-one semester hours are required.

FOCUS OF THIS PAPER

The Industrial Systems Analysis Course (COBOL) has been taught from a textbook without benefit of exposure to the actual equipment. This has been difficult even though every attempt was made to use audio-visual and supplemental textbook materials. In this latter case a supplemental text was adopted (by Dr. Barbay) that presented exercise materials to be woven into classroom presentation. In some offerings, it was possible to visit or use equipment at the base or make an arrangement with a near-by college.

Thus, the background for this development with micro-computers includes an environment that has a military dimension and covers locations remotely scattered all over the United States. The scenario also includes a group of professionals who carry their specialities into those bases on a weekend format where an entire semester's work is covered in three weekends (over six weekends). The pace of delivery is accelerated and brings out the necessity for new tools and techniques.

THE NEED FOR HANDS-ON

We have all long appreciated the need for applying technical training on a device that allows application and response. We also have appreciation for the concept that much of the knowledge learned on one machine is transferable, if the language is one of the more widely utilized ones such as COBOL, FORTRAN, etc., where we have a high level of standardization across manufacturers.

Beyond Hands-on experience, the following were major considerations in moving to micro-computers.

1. Uniformity of the courses delivered in the classroom.
2. Hands-on experience with one of the newer computing devices.
3. Advancing the students into the requirements of a programming language where structure and discipline are characteristics of the language.
4. Adequately documenting the procedures so students can absorb the material presented within a 6 weeks time frame with independent work sessions at the computer.

THE TEACHING OF COBOL AS AN INDUSTRIAL TECHNOLOGY COURSE

In an Industrial Technology program such as ours, COBOL has been the language taught because:

1. It is highly structured and fits into the requirement to teach new comers something about data processing, information systems, computers, etc.
2. It is easily learned and fits the format of a lot of material delivered in a short time.
3. It gives our graduates a language that is the one foremost in use.
4. With COBOL background, BASIC is easy and FORTRAN can be learned where the work environment requires it.

EVALUATION OF EQUIPMENT

In addition to evaluating time sharing approaches, the following equipment was reviewed in selecting the CROMEMCO SYSTEM THREE:

Equipment Evaluated	Model	CPU
1. Southwest Technical Products	6800	6800
2. North Star Horizon	Horizon II	280
3. Ohio Scientific	C-3A	280/ 6502/8080
4. Radio Shack	TRS-80	280
5. Commodore	PET	650L
6. CROMEMCO	System 3	280

Commodore, PET, and Southwest Technical Product's 6800 were eliminated due to the lack of software. (Microsoft requires a 280 CPU for COBOL and FORTRAN). In addition, the College of Engineering has both of these computers and experience there led to the judgement that these machines are unsuitable for the intended application. Personal experience on the machines by committee members facilitated the evaluation process.

The quantity of memory available was considered important to using COBOL. Both the North Star and Radio Shack did not make use of the addressing capability of the 280. North Star addresses 56K of RAM whereas the TRS-80 addresses 48K. Neither the North Star nor the TRS-80 supported COBOL.

Software costs (licensing) were minimized by buying the system from CROMEMCO with software negotiated as part of the price. Since we received a 25% discount on the purchase, our costs were less than if separate software contracts were negotiated.

Software capabilities on the system 3 include:

1. COBOL
2. BASIC
3. FORTRAN
4. RATFOR
5. DATA BASE MANAGEMENT SYSTEM
6. WORD PROCESSOR SYSTEM
7. MACRO ASSEMBLER

It was determined that all machines would be licensed for COBOL and BASIC whereas other software capabilities would be limited to the development machine permanently stationed at the main campus in Carbondale.

In summary, the reasons for taking the approach we did includes:

1. A Micro-Computer was low cost enough to allow 12 systems to be installed, thus providing coverage of all bases with one at central campus for developmental use.
2. Students see and learn to use an up-to-date system which they will encounter in the field.
3. A fixed system design (centrally developed) allows appropriate control of the development with minimized program maintenance costs.
4. The machines are purchased allowing us to control changing of software updates by the manufacturer.
5. Disk storage problems are reduced since low cost diskettes can be purchases and controlled by each student.
6. Hardware Requirements were met:
   a) Durability
   b) Large memory - 64K
   c) Diskette Storage 256K each
   d) Printer capability
   e) CRT

SYSTEM CHECK OUT

We chose to have all systems delivered to central campus where they were thoroughly tested before delivery to the bases. This proved worthwhile because two of the CRT's and CPU's required repairs.

Two of the machines were used to develop the software and documentation support manuals to tailor the equipment to our uses. This consisted of such things as:

1. Writing software routines to automatically perform functions such as Boot, Accessing the Screen Editor, Cataloging of practice sessions, Updating, Cycling old lessons, etc.
2. Documentation consisted of a user's guide which bridged the gap between the voluminous manuals delivered with the equipment and our needs. In courses where there is a high level of intensity (45 hours in 3 weekends), we require key material in as few words as possible.

COST CONSIDERATIONS

The 12 systems were purchased for about $100,000, consisting of:

12 Printers	$20,000
12 Micro Computers	57,000
12 CRT	20,000
Software	3,000
	$100,000

MAINTENANCE CONSIDERATIONS

This is an important consideration because we are spread out across the U.S. and because of the access to some of our locations. We are now gearing up to perform routine maintenance ourselves and have ordered a "spare" machine and allied test equipment to aid in establishing our own capability. This method is being pursued in lieu of maintenance contracts locally because:

1. The desire to keep all machines compatible and to control modifications and changes.
2. The plan to utilize this equipment in other application areas on campus where micro-processor technology is applicable.
3. The uncertainty, at this time, of the capability of the franchised dealers out there that might work on our equipment.
4. The expense of sustaining 12 maintenance contracts.

We consider this one of our developing areas and our plans might change as we progress. For now, we are proceeding cautiously under the above guidelines.

FUTURE DEVELOPMENTS

We have now completed six offerings of the COBOL class utilizing the micro-computers and are very pleased with the response of students and faculty. We are now developing a self-instruction companion course which will allow students to be introduced to the equipment and to step thru an introduction of BASIC.

Beyond this, we plan to weave the equipment into other classes where computational support is required.

SUMMARY

This approach has solved several pressing problems for us and has greatly strengthened the course content. In addition, we have high hopes for the use of this technology in many of our courses; thus better preparing our students for the "real" world.

We would now like to demonstrate the equipment and the User's Guide documentation that has been developed to facilitate its use.

TEACHING SEMANTIC INFORMATION NETWORKS USING MICROCOMPUTERS

Stewart A. Denenberg
Assistant Professor
Department of Computer Science
State University of New York
Plattsburgh, NY 12901

Background

A graduated set of seven microcomputer BASIC programs accompanied by a workbook have been developed to teach the concepts of semantic information networks within the context of a computer science Data Structures course. This paper will describe the contents of the workbook as well as the pedagogical approach of using a microcomputer and a workbook to facilitate a self-directed, interactive approach to computer science education.

Introduction

Semantic information networks were developed as attempts to model the associative way knowledge might be stored and retrieved in the human brain. In this sense it is both a representation of knowledge (a topic in an artificial intelligence course) and a particular type of data structure (a topic that should be, but is not usually covered in a data structures course). What we wish to do here is discuss "semantic nets" from a data structures point of view, and in doing so illustrate their richness, power and utility.

The semantic net representation of knowledge has been described by Quillian [1] and applied by Carbonell [2] in a computer assisted instruction system called "Scholar" which can answer as well as ask questions about a particular knowledge space---geography in this instance. Wexler [3] and Koffman [4] have also designed and implemented similar systems based on semantic nets that are able to interact with student users.

A semantic net can be understood as a declarative (as opposed to a procedural) representation of knowledge consisting of a set of data and the set of relations between them. More precisely, from the data structures view point, we have the more compact symbolism of a weighted digraph consisting of nodes connected by relations which are represented by the weighted, directed edges.

A simple illustration to clarify how a semantic net actually represents knowledge may be useful. Consider the following miniature world consisting of a family of four -- a father named Bob, a mother named Mary and two children, a boy, Billy and a girl, Suzy. A semantic net reprensentation of this knowledge could look like the diagram shown as Figure 1.

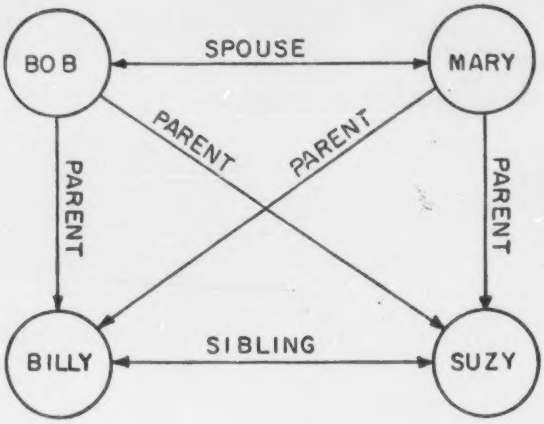


Figure 1

The above semantic net has four nodes (Bob, Mary, Billy, Suzy) and three relations ('spouse', 'parent', 'sibling') that make associations between the nodes. Together the nodes and the relations form a semantic net which is a representation of knowledge about this family.

A richer and more complex set of unidirectional relations could have been chosen to form a semantic net representation of this family. For example, the relation 'spouse' could be replaced by the two relations "is the husband of" and "is the wife of"; the relation 'parent' could be replaced by "is the father of" and "is the mother of"; the relation 'sibling' could be replaced by "is the brother of" and "is the sister of"; for completeness two more relations could be added: "is the son of" and "is the daughter of" resulting in a semantic net as shown in Figure 2.

Either version of the semantic net could be used to represent the hypothetical family; the level of detail depends on how we wish to use the representation as well as on economic constraints. The important point made in the workbook is that there is more than one way to represent knowledge using a semantic net and that there can be more than one data representation for a semantic net when it is implemented in a computer program.

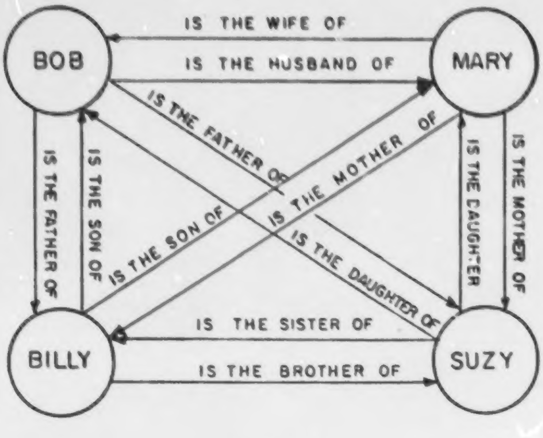
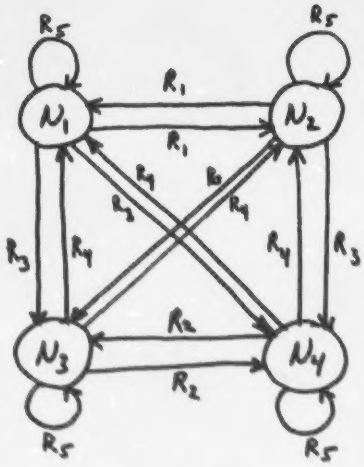


Figure 2

Using the notation 'N' for node and 'R' for relation we can redraw Figure 2 as Figure 3. In the first program, SN1 (for semantic net 1) the student is shown a simple matrix data structure for the semantic net shown in Figure 3 (note that R_5 is not absolutely necessary but is used to simplify the data structure to be developed).



R_1 = 'spouse of'
R_2 = 'sibling of'
R_3 = 'parent of'
R_4 = 'child of'
R_5 = 'same as'

N_1 = 'Bob'
N_2 = 'Mary'
N_3 = 'Billy'
N_4 = 'Suzy'

Figure 3

The student reader is shown that if we use the subscripts of each relation as elements in an interconnection matrix, we have a simple data structure that holds all the information shown in Figure 3.

	Bob	Mary	Billy	Suzy
Bob	5	1	3	3
Mary	1	5	3	3
Billy	4	4	5	2
Suzy	4	4	2	5

Reading across the first row we see that Bob is related to Bob via Relation 5 (R_5 = 'same as'), Bob is related to Mary via R_1 (spouse of), and Bob is related to both Billy and Suzy via R_3 (parent of). Similar information is gained by reading across the Mary, Billy and Suzy (rows 2, 3, and 4) rows of the matrix. Thus, with the appropriate interpretation of its elements, the following matrix:

$$\begin{pmatrix} 5 & 1 & 3 & 3 \\ 1 & 5 & 3 & 3 \\ 4 & 4 & 5 & 2 \\ 4 & 4 & 2 & 5 \end{pmatrix}$$

along with the characters vectors R and N, as defined in Figure 3, constitute a data structure representation of the semantic net also shown in Figure 3.

Next the student is invited to write a BASIC program that performs a simple inquiry function on the suggested data structure: the input is to be any two nodes in the network and the output is to be the relation between those nodes. For example, if the input were (BOB, BILLY) coded as (1,3) since they are nodes N_1 and N_3 respectively, the output from the program should be:

BOB IS THE PARENT OF BILLY

The student is invited to try writing SN1 himself before utilizing the version on the microcomputer system and is encouraged to peruse the listing of the program which appears as an appendix to the workbook as well as try to improve the program by making changes to it. This pedagogy is repeated for each of the other programs SN2 through SN7.

Simple Inquiry Functions, More Complex Data Structures

Programs SN2, SN3, and SN4 embody alternative data structures to handle a more complex family situation. Suppose we add the following knowledge to our semantic net:

Bob is older and taller than anyone else in the family.

Mary is older and taller than Billy and Suzy.

Billy is older than Suzy.

Suzy is taller than Billy.

We can incorporate this knowledge into the semantic net of Figure 3 by expanding the set of relations to

R_1 = 'spouse of'
R_2 = 'sibling of'
R_3 = 'parent of'
R_4 = 'child of'
R_5 = 'same as'
R_6 = 'older than'
R_7 = 'taller than'

thus leading to an interconnection matrix with variable length vectors rather than scalars as elements:

$$\begin{pmatrix} 5 & 1,6,7 & 3,6,7 & 3,6,7 \\ 1 & 5 & 3,6,7 & 3,6,7 \\ 4 & 4 & 5 & 2,6 \\ 4 & 4 & 2,7 & 5 \end{pmatrix}$$

We now have a more complex network where two nodes can have many relations (going in the same direction) between them.

Since most high level programming languages (including BASIC) can not handle matrices whose elements are variable length vectors (sometimes called ragged arrays) the student is advised that he must supply the software necessary to implement this more complex data structure. Programs SN2, SN3 and SN4 lead the student through a detailed discussion of how multidimensional arrays and how single and multiple level directories can be used to implement the simple inquiry function of SN1, namely: given two nodes, print out all of the relations between them. For example, if the input is specified as 1,2 (the subscripts of nodes N_1 and N_2 which denote Bob and Mary respectively) the output from SN2, SN3 or SN4 would be:

```
BOB IS THE SPOUSE OF MARY
BOB IS OLDER THAN MARY
BOB IS TALLER THAN MARY
```

Intermediate Inquiry Functions, More Complex Programs

A semantic network is a representation of knowledge and programs SN1 - SN4 are tools one can use to search thru knowledge. In a sense we can view the nodes and relations of a semantic net as a "knowledge space"; our programs are search tools that facilitate acquisition of this knowledge and illuminate the structure of this knowledge space. A particularly useful search tool would be the facility to view all of the nodes one relation away from a chosen node. Often the view of the total semantic net is "too cluttered", i.e., there is too much information present to gain any insight or understanding of the knowledge space. However, imagine that the entire semantic net resides in darkness and that we have a movable spotlight that can illuminate any given node and its outpointer relations to a neighborhood extending exactly one node away. Now we are viewing only a small portion of the network at any one time which effectively reduces the amount of information presented.

Suppose, for example, we wish to see how Mary is related to the rest of the members of the semantic net as defined in Figure 3. Graphically this is equivalent to placing Mary's node at the center of the network and shining a spotlight that shows Mary's node and only those nodes exactly one relation away as depicted by Figure 4.

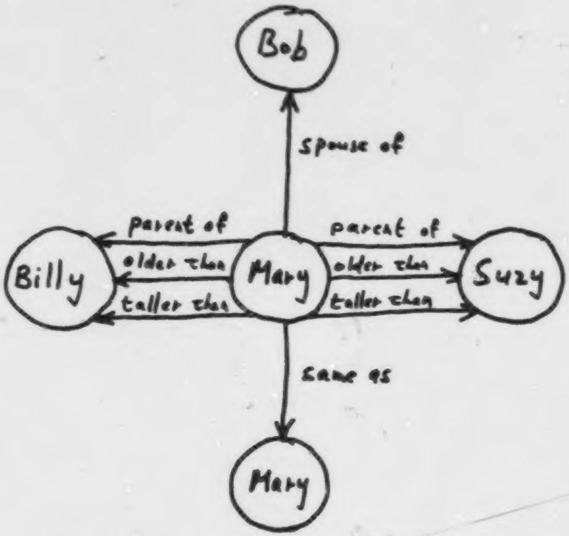


Figure 4

The student is shown that the same data structure used in SN4 can be extended to create SN5, the "spotlight" program. The input to SN5 is the central node of the spotlight and the output is a list of all nodes in the net exactly one relation away. For example, if the input were the code 2 (for Mary who is N_2) then 8 lines of output would be produced:

```
MARY IS THE SPOUSE OF BOB
etc.
```

Intermediate Inquiry Functions, More Complex Programs, More Complex Data Structures

Programs SN6 and SN7 perform the same inquiry functions as SN2, SN3 and SN4 using more complex programs and different data structures called node-tables and relation-tables.

The node-table data structure is derived from the 'spotlight' function of SN5: we have a central node and a set of nodes one relation away. This can be represented by a two-column matrix (which we call a table) whose name is the central node; the left column holds the subscripts of the relations and the right column the subscripts of the nodes the relation points to; for example:

N_1

R	N
1	2
1	3
2	4
2	5
3	2

represents the situation that node N_1 has five outpointer relations: R_1 which points to N_2 (see the first now in the table), R_1 which points to N_3, R_2 which points to N_4, R_2 which points to N_5 and R_3 which points to N_2. Graphically this looks like Figure 5.

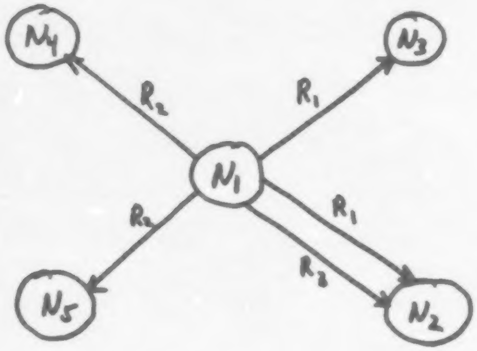


Figure 5

The tables for each node can be stacked into one long table and a simple one-level directory can be used to discriminate between each component node-table. SN6 is the program which uses this data structure to implement the inquiry function described in programs SN2 through SN4. Program SN7 does essentially the same thing as SN6 but inverts the node-table format to a relation-table format wherein the name of each table is referenced by the subscript of a relation and the entries specify the subscript of the node at the tail of the relation and the subscript of the node at the head of the relation.

Distributed throughout the discussion of programs SN2 through SN7 are comments on the relative advantages and disadvantages of the type of data structure chosen with regard to storage versus speed considerations. The student's understanding that a savings in memory storage is usually paid for at the price of increased execution time of the program (and vice-versa) is reinforced.

Also discussed is the tradeoff between the update and interrogate (inquiry) functions. Programs SN2 to SN7 attempt to show that a data structure that facilitates a particular inquiry function makes updating of that data structure (when nodes and relations must be added or deleted from the network) more difficult. Conversely, a data structure that is easier to update results in more complex logic in the inquiry program.

Advanced Inquiry Functions

The penultimate section of the workbook explores other possible useful inquiry functions over a semantic net. Along with examples, the following eight inquiry functions are suggested:

1. Input: 1 node, 1 relation
   Output: All different nodes connected to input node by input relation.

2. Input: 2 nodes
   Output: All relations between the 2 input nodes.

3. Input: 1 node
   Output: All outpointer relations and the nodes they point to.

4. Input: 1 node
   Output: All inpointer relations and the nodes they are outpointers from.

5. Input: 1 relation
   Output: All ordered pairs of nodes connected by that relation.

6. Input: 1 node
   Output: All paths in the semantic net that terminate in that node.

7. Input: 2 nodes
   Output: All paths from the first to the second specified node.

8. Input: 2 nodes, R relations
   Output: All paths between the 2 input nodes that can be generated using only the given R relations (R can be 1).

Of course the student is invited to think of other possible inquiry functions and the data structures and algorithms necessary for their implementation. The student is encouraged to relate the data structure/algorithm system (a computer program) to speed vs storage and update versus interrogate functions.

Selected Student Projects

Eight student projects are described in detail in the workbook: Family relations, Course offerings in a college catalog, Car pooling, Systems analysis information flowcharts, Finite state automata, Dictionary, Furniture world, and A generalized system to create and manipulate semantic networks. Due to space limitations only one will be described here; readers interested in the other examples or in a more detailed discussion of the programs may request a copy of the workbook from the author.

Course Offerings in a College Catalog

Students and their advisors are constantly trying to put together a program of courses that satisfy sometimes conflicting requirements. There is the set of courses in the Major requirement (what the faculty thinks the student should know for depth of knowledge); the General Education requirement (what the faculty thinks society thinks the student should know---for breadth of knowledge); and the Elective requirement (what the student thinks the student should know). While the Major requirement is usually fixed and somewhat inflexible, the General Education and Elective requirements can usually be fulfilled by many alternative paths. If we select the General Education program as an example, we could set up a semantic network of courses and the relations between them so that students and advisors can choose a path that integrates the knowledge gained under a set of specified relations.

We can view each course in the curriculum as a node in a semantic net and the major themes that integrate the courses as relations. For example major integrative themes might be 'war', 'marriage', 'love', 'power', etc. Thus a semantic net representation of a portion of the College Catalog might look like Figure 6 (W = 'war', etc.).

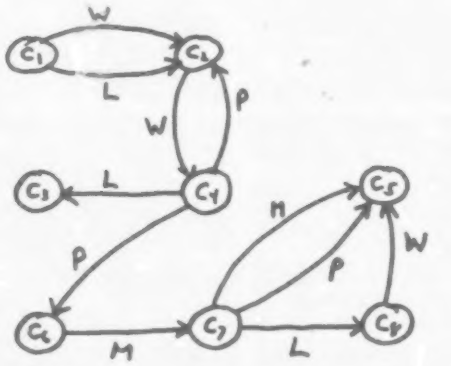


Figure 6

The student is invited to create a semantic network of the College Catalog that can be interrogated through one or more of the eight previously listed inquiry functions.

Conclusion

The workbook and the BASIC programs which are available on a Heathkit H8/H17 microcomputer system constitute a self-directed, interactive approach to a special topic of computer science education: semantic information networks and their data structures. Currently the workbook is being evaluated by senior undergraduate computer science majors at SUNY, Plattsburgh. Based on their comments and evaluation, the workbook will be revised and incorporated into the Data Structures course in the Fall of 1980. A more formal evaluation of the educational efficacy of this approach will be made at that time.

References

1. Quillian, M.R., "Semantic Memory" in Semantic Information Processing, editor Marvin Minsky, (Cambridge, Mass: MIT Press, 1968).
2. Carbonell, J., "AI in CAI: An AI Approach to CAI," IEEE Transactions on Man-Machine Systems, Vol. MMS-11, No. 4, Dec., 1970, pp. 190-202.
3. Wexler, J.D., "Information Networks in Generative Computer Assisted Instruction," IEEE Transactions on Man-Machine Systems, Vol. MMS-11, No. 4, Dec., 1970, pp. 181-189.
4. Koffman, E.B. and Blount, S.E., "Artificial Intelligence and Automatic Programming in CAI," Third Joint International Conference on AI, Stanford, Calif., Aug., 1973, pp. 86-94.

HUMANISTIC COMPUTER PROGRAMMING: A GATEWAY TO LEARNING

George Donoian, Ed.D.
Business Administrator
Detroit Public Schools

Taxpayers are demanding results from schools. Current accountability efforts have seized upon computers with early gains indicating student success. The author warns this cognitive mastery is not sufficient. Machine teaching does not provide for affective education. Therefore, he proposes humanistic content for computer programs as the gateway to continued learning.

INTRODUCTION: EDUCATION BEARING THE BRUNT OF TAXPAYERS' REVOLT

While American history abounds with revolt, the most compelling force traditionally angering us is the matter of taxation. From the start, we Americans have rebelled against tyrannical efforts to squeeze more tribute out of us. This is still true today. Witness how Proposition 13 has swept eastward from California like an uncontrolled fire.

More directly, the issue of school taxation leads an angry constituency to demand some form of accountability while they understandably and reluctantly expect educational costs to rise like everything else. Property owners, and particularly parents, want results for the salaries paid to teachers who consider themselves professionals and are expected to be paid such salaries. Just where are the corresponding results? If teachers teach, then teaching should lead to learning. Not so, says the public emphatically! Whatever some teachers are doing, it is not teaching, because too many students cannot do simple math, write legibly or read effectively. Worse yet, some students do not show any hint of having been to school at all, cite angry parents at board meetings. As harsh as this sounds, the sad truth is that some students do fit these descriptions. Of course, the public perception and feelings are real and pervasive. These "facts" alone are enough to arouse the already dismayed public and more importantly, to do something about it. Thus, battle lines are drawn and urban parents in particular are most explicit about their expectations of educators.

EDUCATIONAL RESPONSE TO PUBLIC CRITICISM

*As an urban educator for over twenty-five years, I have been both a witness to and participant in the educational process. In that time, I have seen the rise and fall of many tides upon which American education has ridden--accountability as the most current wave has brought many panacea that include teaching by objectives, objective referenced testing, computer-assisted and managed instruction and a plethora of trendy notions which like other past responses such as classroom TV, team teaching, ability grouping, compensatory education, etc., will hold off the public for a while. These catchy "innovations" have stalled the dissatisfied public in the past, but I believe this time it is different-the public will not be thwarted easily by our new crop of "program priorities". For example, the current movement calling for high school competency skills expects students to demonstrate that twelve years of attendance and hopefully, education, have had an effect. Students should not only claim that their long tenure under the tutelage of teachers has had a definite impact upon them, but also demonstrate it with fact.**

SCHOOLS EMPHASIZING COGNITIVE TEACHING

There has been a definite payoff by focusing upon cognitive teaching and learning. Let us not rejoice so quickly with our initial test results. There is ample research evidence to demonstrate that regression can and will set in when our parents, teachers and students get tired of the heavy stress upon cognitive skills and knowledge acquisition at the neglect of the affective learning domain.

One need only to look at computer-based education to see how thoroughly cognitive the curriculum is. This situation is problematic because school districts have defensively embraced the cognitive realm with the near absence of anything affective. We admit that computer math and science courses are certainly intriguing and hold sway over some youth who enjoy the novelty of a machine and mastery over new material presentation. Let this heavy dosage of cognitive learning continue and become intensified, then ask seniors about

**Schools in the State of Virginia will be issuing four different high school diplomas to graduating seniors based on their tested competency.*

their attitudes toward sheer "facts learning". Once invited to express their honest feelings without reprisals, our graduates will tell us startling tales about the shortcomings of their education.* Without "picking on" computer education, science or mathematics teachers specifically, we will discover the long-term effects of these exact sciences being stressed again and again. When standards of performance take on the greatest importance and when achievement is not regarded as a process, but more precisely as a product with its own intrinsic value, then education loses its relevance and raises itself above all. In this exalted position, cognitive learning becomes an untouchable, unrewardingly demanding and rigidly inhuman, lacking sensitivity to student needs and perceptions.

ANALYSIS OF CURRENT CURRICULAR PROBLEMS

In the headlong pursuit of achievement for today's youth, teachers have almost forgotten to ask some fundamental questions about the content and methods which have recently been introduced into schools. The quest for cognitive mastery has not been all that bad, but it has left some questions unanswered and created a great deal of uneasiness about the so-called success of the achievement orientation along with the computer approach to learning. One of the pressing issues has been that of relevancy. What materials and what methods seem to produce the best results? What values do they engender and for whom?

In reaching for answers, too many of our colleagues have lost sight of student needs, wants, wishes and interests. Thus the system is the end and not the means for accomplishing the mission. Since cognitive teaching lends itself to easy description and machine manipulations, we find that with computer-based education, for example, a general absence of values clarification, humanistic touches and other affective thema, because it is too heavily loaded with subject matter. On the other hand, the philosophers among us could hold up progress with incessant questions, preventing us from accomplishing our goals. We must plan for pondering possible meanings, values and ultimate significances but we must also get the job done, which is teaching for learning, learning for achievement and achievement for education. Without both cognitive and affective learning, education becomes sterile and weak.

Another moot issue modern teachers raise is the matter of necessary skills and concepts. When all is said and done, education is a process of refining students, of increasing their sophistication, of explaining to them the complex in understandable terms, of having them explore the unknown and converting it into the known, of their making use of what seems isolated into an integrated whole and of filling in their information gaps with knowledge.

Further complicating education with its emphasis upon cognitive learning is the matter of curricular linearity, i.e., the relationship of skills and knowledge as they affect the students' future educational needs. While the responsibility is placed squarely upon the teachers now for the courses to be taught, how do these studies contribute to and prepare students? Or are we guilty of being protectionistic and paternalistic about our own survival? Dare we let the chips fall where they may for next year and thereafter? What we are stressing today should have a direct bearing upon students' future. Why shouldn't there be continuity and relevance in the curriculum? Why shouldn't there be some synergistic effects in our curricular offerings so that students can be expected to grow, say, to solve complex sciences and mathematics by means of the computer or to expand their social studies horizons beyond their immediate neighborhoods so that they will be able to analyze and solve global problems, after all we profess about brotherhood of mankind?

Today's education is so burdened with its task to demonstrate how viable and accessible it is that we fail to notice the uneven path upon which we are directed to go by being reactive to the public. What we educators should be concerned about and working diligently for are the minimal expectations of any well-formulated curriculum as characterized by the following:

1. That the content is relevant to the times and for the students with a perspective of the past and design for the future.
2. That today's learning experiences contribute to students' education for the coming years.

Henry Thoreau expressed it succinctly when he spoke of the values of a humanistic education:

> "It is something to be able to paint a picture, or to carve a statue, and so to make a few objects beautiful. But it is far more glorious to carve and paint the atmosphere in which we work, to affect the quality of the day--this is the highest of the arts."*

What is proposed must have the hallmarks of a humanistic education if we are to produce good students and citizens who will be able to function effectively as the future generation. Merely providing transitory school achievement opportunities cannot and will not produce humanitarian qualities and values in our students who must survive and hopefully, thrive in a world environment with shrinking resources, international hostilities, economic upheavals and advancing technologies. We educators have the major obligation of causing great changes. I refer to the first Renaissance which hastened mankind out of the Dark Ages and thereby quickened the need for knowledge, freedom and personal fulfillment. Indeed, the process began modestly and almost inauspiciously with King John signing the Magna Carta on the plains of Runnymede in 1215 A.D. Surely this event had many predisposing events and many yet to come before human rights as we enjoy them became realities in the United States. Bloodshed, wars and destruction of property have been the ultimate price tags for establishing and guaranteeing the inalienable rights we cherish while other people around the world have yet to achieve. Although we have earned ours by difficult means, we stand to lose some if each generation does not reaffirm these freedoms. This is accomplished by teaching values to our students and thereby transmitting our political and personal heritage. Education should be at the forefront of that movement safeguarding those rights for the coming generations. We do not need another million years of struggle to discover the efficacy of an educative process as the means of improving mankind and the wisdom to do something about it. We must act now to safeguard the humanism we inherited from the past.

---

*See Carl G. Jung's Memories, Dreams, Reflections, edited by A. Jaffe, Random House, New York, 1965.

*P. 81, Henry Thoreau, Walden, Modern Library, New York, 1965.

IF WARNING IS UNHEEDED

I forecast that if the present trend continues with its stress on cognitive teaching as exemplified by computer-based instruction, we will logically and inevitably cause our students to reject and rebel against our defensive measures to redeem ourselves as educators. As we continue to pressurize our students to achieve and thereby justify our presence in schools, we will pay dearly for having been so "unbalanced" in our curriculum. Without any humanism to offset the heavy dosage of cognitive teaching, our students will be denied fundamental skills, knowledge and attitudes which could help them to be better citizens, especially as we look to the next decade with the burgeoning use of computers taking over human effort.* Depersonalization of society may be inevitable, but our schools should not be accelerating nor promoting this process by training the next generation of adults to accept, enjoy and function without affective substance. Today's education stands at this threshold of responsibility and in what ways we respond and fulfill our mission will affect the whole fabric of public education in America as we close this century.

THE RENAISSANCE MOTIF

What must we do to bring about a Renaissance? How will we recognize a rebirth of this spirit in American schools? The writer proposes a humanistic education if we are to turn out good students and good citizens who will be functioning effectively in a world of problems seeking answers. For tomorrow's computer-assisted and -managed education to be humane, we need more than a flurry of activity, i.e., (teaching) students to achieve cognitively. That is simply quantitative effort. An achievement campaign is only the opening thrust of a Renaissance. Our Renaissance calls for much more. Our humanistic motif requires an imaginative response from us to broaden our students' horizons by increased learning experiences which introduce and enhance their literary, artistic and cultural lives and not at the neglect of their scientific knowledge, if our students are to become effective scholars.

We cannot equate the previous Renaissance to the one which, we believe, is upon us now. In that respect we certainly hope that this one does not require spilling of blood and all that defies any semblance of humanity in the face of all the wars, oppression and tyranny that man painfully experienced since the first enlightenment period. On the contrary, too many predisposing factors prevail today which cannot return us to the Dark Ages to start anew.

HUMANISM FOR A COMPUTER-ORIENTED EDUCATION

In addition to our conviction and commitment that our students can learn, we educators must provide a greatly expanded curriculum which will be the basis for a humanistic education. In short, the quantitative aspect of our proposed teachings will require more resources and more imaginative use of the ones we already have. Further, by increasing learning opportunities, we will be introducing our students to more than a few textbooks as the curriculum of humanism.

More significant than the extent of our offerings is the matter of our humanistic educational philosophy. What must we do to be certain that our humanistic motives permeate and saturate the curriculum? How do we guarantee a humanistic education? I believe that the following six requirements for a humanistic computer-oriented education are essential criteria:

First, the curriculum, the school, staff and society must serve the centrality of every human being and his self-development by providing for students' active participation in the design and direction of the curriculum. For as long as students are prevented from making some choices, then they are relegated to a passive mode of learning. Because learning specialists seek an active involvement on the part of students as learners, then we must bravely provide our charges with endless opportunities to develop and practice their skills in intellectual exploration, random discoveries, individual analysis and synthesis. They should be taught to think critically and master study skills that will help them arrive at their own conclusions and solutions, and furthermore, have the desire and courage to tackle new problems more difficult than the previous ones.

Second, we must operate with the imperative that as humanistic software people and teachers we cherish students' purposes and perceptions equally as our own. We must give up our long-held beliefs that we know better than they about what may be good for them to know. Perhaps it has been this perception which has condemned students as having "lack of motivation". What if we turned it around and allowed our students to study the content whit they themselves see fit for their purposes? We would witness a quantum increase in motivation and learning--a veritable learning revolution with intrinsic value for students rather than offering them external goodies, such as good marks.

---

*See 17th Annual A.E.D.S. Conference Proceedings, "Humanism for the Computer Age In Education: Planning for Man/Machine Synchronicity", George Donoian, pp 321-24, 1979.

Third, just as it was for the first Renaissance, infallibility and dogma must give way and be replaced by variability and tentativeness of knowledge in the humanistic curriculum for our scholars. Their logical inquiry and defense of any issue will depend upon the skills we teach them. How much more learning can take place when our students are skilled in dialectics and debate? Developing student argumentation skills in lively mock courts and other oral defense sessions would be useful and necessary.

Fourth, our humanism must reach beyond the narrow confines of our own time parameters. Truly our humanism must embrace a time management which keeps the students' own purposes and perceptions at the very forefront of their education, not our own directives. What we perceive as adequate learning time may not be so to our students who are the learners. Therefore, program writers and teachers must be ready to dwell upon a unit of study longer than we had planned because the learners may be enraptured by it or that students feel it has not been adequately dealt with in depth, thereby requiring several more weeks of study.

Fifth, in keeping with students' purposes, perceptions and time management, we ought to be certain that their personal wisdom is respected and honored, since we already know student excursions into the world of self-directed learning are the most poignant ones. And yet, today we discourage, penalize and prevent students from doing just that because they deviate from our daily plans and semester designs to cover a curriculum that is usually sterile, deadly and, most likely, useless.* Instead, we as humanistic educators should increase our efforts by sending our students onto wide-ranging cerebral trips seeking the obscure, the trivial and "useless" bits of information which will provide them with significant knowledge, answers to difficult questions and solutions to challenging problems. Learning must be fun for the learner.

And sixth, the value of an humanistic education has got to be in sharing it with one's fellows. Vocationally, too, our students will measure its utility they make of their humanism and its ego-enhancing qualities in the marketplace, not the least of which is the satisfaction for a lifetime of service to one's self and others, be it in the arts or sciences.

CONCLUSION

I believe that today's education is at a crossroads where it must decide its own future, or forego any hope of being its own master. It can't continue to make excuses, rationalized with feeble and catchy reasons and terminology about how it is attempting to overcome problems. One such way out of this dilemma is to be preoccupied with teaching for the sake of achievement - only a step out of the quagmire. However, this effort is insufficient to propel education toward humanism.

I have proposed that the computer can be a powerful tool of the second Renaissance by embracing a philosophic outlook with the following six parameters for teaching and learning:

1. That every student is involved actively as a learner,
2. That each student's learning purposes and perception be respected,
3. That variability, not infallibility, dictates our teaching designs and hence our students' learnings,
4. That the students' time management is more relevant than the teachers' own chronologic impositions,
5. That student judgement and wisdom should prevail over the teachers' when we are determining the what, how and why of learning, and finally,
6. That the ultimate value of a humanistic education is in its use for self-enhancement and for work as service to oneself and others employing a foundation of learning in both the arts and sciences.

We will witness a new Renaissance when our schools operate with these six characteristics as the epicenter of a humanistic education. A greatly expanded curriculum will permit our students to have freedom of choice, personal dignity, and increased knowledge and skills befitting enlightened scholars who are equally at home in the arts, as well as the sciences. Truly they will be modern-day Renaissance people.

---

*Jung, op. cit., pp. 24-83.

COMPUTER SCIENCE EDUCATION IN SMALL COLLEGES: THE CURRENT STATUS

Gerald L. Engel
Christopher Newport College

Abstract

This paper presents the results of a study of computer related bachelor degree programs at institutions enrolling less than 2500 students and at institutions enrolling 2500-9999 students. Areas which are investigated include number of programs, enrollments, types of programs, staff size, and course information. Results are compared between these two classes of institutions, and with similar data for all institutions offering such programs.

Introduction

Computer education at smaller institutions has been the subject of a number of papers in recent years. Lopez, Raymond and Tardiff [1] surveyed the offerings in small liberal arts colleges enrolling 1000-3000 students. Discussion of specific programs are given by Ahlgren, Sapega, and Warner [2], Lopez [3], Worland [4], Winrich and Peterson [5], Cameran and Karian [6], and Smith [7]. A philosophic argument regarding computer science in liberal arts institutions is given by Mein [8]. Earlier reference to computing in small colleges may be found in the survey of post "Curriculum '68" literature in Computer Science Education [9].

In several of the papers cited above, as well as informally, there have been calls for curriculum guidelines directed to the smaller schools. Prerequisite to the development of such guidelines it is necessary to determine the current status of Computer Science in such institutions. This paper reports the results of such a survey. The Fourth Inventory of Computers in Higher Education [10] provides the data for the survey. An interpretive report has been prepared which overviews programs at the bachelor degree level at all institution based on these data [11]. In this paper special focus will be given to institutions enrolling less than 2500 students and institutions enrolling 2500-9999 students. For comparison purposes data will also be used from the 1969-70 Inventory of Computers in Higher Education [12].

Status of Programs

Table 1 presents the estimated number of computer related bachelor degree programs broken down by size of institution and type of support. The institutions in these size classes, offering the bachelor degree, that responded to the survey are listed in Appendix A.

Table 1
Estimated Number of Bachelor Degree Programs by Size of Institutions and Type of Support

	No. 69-70	No. 76-77	% Inc.
All Institutions	198	541	173
All Institutions below 10,000	122	321	163
Private	49	159	224
Public	73	162	122
All Institutions 2500-9999	82	225	174
Private	25	80	220
Public	57	145	154
All Institutions 500-2499	40	96	140
Private	24	79	229
Public	16	17	6

The growth in the number of degree programs is most striking, especially the major growth which is seen in private institutions enrolling less than 10,000 students.

Table 2 further breaks down the number of programs in institutions enrolling less than 10,000 students, by dividing them into classes by the types of degrees which are awarded.

Table 2
Estimated Number of Bachelor Programs by Degrees Offered and Size and Type of Institution

	Bachelor Only		Bachelor & Associate		Bachelor & Graduate		Bachelor Assoc.&Grad	
	69-70	76-77	69-70	76-77	69-70	76-77	69-70	76-77
All Institutions below 10,000	73	220	18	54	30	43	1	4
Private	29	100	10	28	10	29	0	2
Public	44	120	8	26	20	14	1	2
All Institutions 2500-9999	41	150	14	33	26	38	1	4
Private	9	40	7	12	9	26	0	2
Public	32	110	7	21	17	12	1	2
All Institutions 500-2499	32	70	4	2.	4	5	0	0
Private	20	60	3	16	1	3	0	0
Public	12	10	1	5	3	2	0	0

From this table it can be seen that the majority of programs both in 1969-70 and in 1976-77 were bachelor degree only, and this was even more evident in the case of institutions enrolling 500-2499 students. It should be noted that the class of institutions enrolling less than 500 contributed no programs to the total in either 1969-70 or in 1976-77.

The distribution of program titles corresponded closely to the distribution for all institutions as is seen in table 3.

Table 3
Distribution of Programs by Program Title
1976-77

Program Name	% of Programs at institutions less than 2500	% of Programs at institutions 2500-9999	% of Programs at all Institutions
Computer Programming	2	2	1
Computer Science	54	56	58
Computer Science Technology	2	1	1
Data Processing	11	12	9
Information and Computer Sci	2	2	2
Information Science	5	[illegible]	3
Information Systems	2	8	8
Mathematical Sciences	2	0	0
Other	20	17	18

Of some note is the fact that programs in data processing showed a higher percentage in the two classes of smaller institutions. It is also of note that the programs in computer engineering, computer and information science, and computer science and engineering had no offerings in either of the two classes of institutions, while accounting for more than 1% of the programs in each case in the category of all institutions.

Table 4 presents the estimated number of undergraduate students in both years surveyed, and the number of graduating students in those years.

Table 4
Estimated Number of Undergraduate Majors and Graduating Students

	No. of Undergraduate Majors		No. of Graduating Students	
	69-70	76-77	69-70	76-77
Institutions 500-2499	1820	3136	40	145
Institutions 2500-9999	7626	18510	1084	2950
All Institutions	65916	105190	2157	9517

From this table it may be seen that while the number of programs at institutions enrolling below 2500 accounted for 18% of all programs, these institutions only accounted for 3% of the undergraduate majors in 1976-77 and 2% of the number of graduating students. Institutions enrolling 2500-9999 students accounted for 42% of all programs in 1976-77 but accounted for only 18% of the undergraduate majors in this academic year and 31% of the graduating students. This implies that the programs at the institutions are, in fact, relatively small. It is of interest to note that the increase in the number of graduates at schools in the 500-2499 is a higher percentage than the increase in the number of programs.

Table 5 considers the staffing of programs.

Table 5
Staffing at Institutions Offering at most the Bachelor Degree and at All Institutions 1976-77

	FTE/ Program	Ph.D/ Program	Ph.D/CS per program
Below 2500	2.3	1.1	0.1
2500-9999	3.5	1.0	0.3
All Institutions	6.6	2.1	1.4

The staffing difficulty of computer related degree programs, especially as it applies to staffing programs with individuals with a Ph.D. in Computer Science, is well illustrated by this table. The table also illustrates that the problems of the small college regarding staffing are quite severe, and that the average FTE/program falls below the four specified in the accreditation guidelines for undergraduate programs in Computer Science specified by the Association for Computing Machinery [13].

Status of Courses

Appendix B presents estimated course enrollments for computer related courses at institutions offering at most the bachelor degree and enrolling below 2500 students, and those enrolling between 2500 and 999 students, as well as all institutions for academic year 1976-77. The total for the institutions enrolling less than 2500 is 4.6% of the total for all institutions. Courses which showed a higher percentage enrollment compared to all institutions include the following:

Computer Appreciation
Introductory Course in a Computer Language
  like FORTRAN, PL/I, BASIC or ALGOL
Introductory Course in a Language like COBOL
Problem Solving Course in Engineering or
  Scientific Applications
Advanced Course in Programming
Sequential Machines
Advanced Numerical Analysis
Advanced Operations Research
Data Preparation
Unit Record
Information Systems Design (Advanced)
Ecinsmetric Simulation
Instructional Design

In almost all cases, these courses represent service offerings or application areas, pointing to more emphasis in these areas in the institutions enrolling less than 2500 students.

Considering institutions enrolling 2500 to 9999 students, the total enrollment for all courses is 3.4% of the total for all institutions. Courses which showed a higher percentage enrollment than that for all institutions include the following:

Introductory Course in a Language like COBOL
Introductory Course in Programming
Artifical Intelligence
Computer Organization
Advanced Computer Organization
Introductory Numerical Analysis

Data Preparation
Unit Record
Introductory Course in Data Processing
Advanced Course in Data Processing
Introductory Systems Analysis
Large Scale Information Processing Systems
File Organization
Information Systems Design (Introductory)
Information Systems Design (Advanced)
Management Information Systems
Management of Computer Centers and/or
  Information Processing Departments

The selection of courses which are shown here points to an emphasis in the areas of data processing and information systems at institutions with enrollments in the range 2500-9999.

Summary

The development of computer related academic degree programs at the bachelor level has been significant both in the class of institutions enrolling less than 2500 students and those enrolling between 2500 and 9999 students. The growth in the number of programs has been most significant in privately supported institutions.

Both the class of institutions enrolling less than 2500 students and those enrolling 2500-9999 students share a number of things in common. The bachelor degree program is primarily the only one offered, the programs are small in comparison to all institutions and staffing represents a major problem area.

Though the titles by which the programs are identified are approximately the same as those offered in all institutions some differences exist in the type of courses which are offered. In the case of the institutions enrolling less than 2500 students there is more emphasis on courses in application areas and on service courses. In the case of the institutions enrolling 2500 to 9999 students there is more emphasis in course in the areas of data processing and information systems.

References

[1] A.A. Lopez, R. Raymond, and R. Tardiff. "A Survey of Computer Science Offerings in Small Liberal Arts Colleges." *Communications of the ACM* 20, 12 (December 1977), 902-906.

[2] D. Ahlgren, A. Sapega, and H. Warner. "A Sequence of Computing Courses for Liberal Arts Colleges," *SIGCSE Bulletin* 10, 1 (February 1978), 180-182.

[3] A. M. Lopez, "An Implementation of ACM Curriculum 77." *SIGCSE Bulletin* 10, 2 (June 1978), 47-52.

[4] P. B. Worland. "Using the ACM Computer Science Curriculum Recommendations in a Liberal Arts College." *SIGCSE Bulletin* 10,4 (December 1978), 16-19.

[5] L.B. Winrich and C.G. Petersen. "A Pragmatic Implementation of Curriculum '77 at the University of Wisconsin-LaCrosse." *SIGCSE Bulletin* 10, 4 (December 1978), 37-41.

[6] J.S. Cameron and Z.A. Karian, "Computer Science Curricula for Small Colleges." *SIGCSE Bulletin* 11, 1 (February 1979), 215-219.

[7] J. Smith. "The Small Liberal Arts College: A Challenge for Computer Science." *SIGCSE Bulletin* 11, 1 (February 1979), 220-223.

[8] W.J. Mein. "Computer Arts: Computer Science in a Liberal Arts Environment." *SIGCSE Bulletin* 11, 1 (February 1979), 224-227.

[9] R.H. Austing, B.H. Barnes and G.L. Engel. "A Survey of the Literature of Computer Science Education Since Curriculum '68." *Communications of the ACM* 20, 1 (January 1977), 13-21.

[10] J.W. Hamblen and T.B. Baird. *Fourth Inventory of Computers in U.S.* Higher Education, 1976-77. EDUCOM, Princeton, N.J. 1979.

[11] G.L. Engel. "Computer Science Related Degree Programs at the Bachelor's Level." in *Fourth Inventory of Computers in U.S. Higher Education*, 1976-77, Interpretive Reports, EDUCM, Princeton, N.J. 1980.

[12] J.W. Hamblen. *Inventory of Computing in U.S. Higher Education 1969-70: Utilization and Related Degree Programs*. U.S. Government Printing Office, Washington, D.C.

[13] Accreditation Committee, Association for Computing Machinery. "Accreditation Guidelines for Bachelor's Degree Programs in Computer Science." *Communications of the ACM* 20, 11 (November 1977) 891-892.

Appendix A - Responding Institutions

1) Public institutions enrolling less than 2500 reporting a computer related bachelor degree program being offered in 1976-77.

Alabama
  Livingston University

Arkansas
  Arkansas Technical University

Connecticut
  U.S. Coast Guard Academy

Kentucky
  Kentucky State University

New Mexico
  New Mexico Institute of Mining and Technology

Ohio
  Central State University

Oregon
  Oregon Institute of Technology

Pennsylvania
  Lock Haven State College

Texas
  Texas A&I University, Corpus Cristi

West Virginia
  Shepherd College

2) Public institutions enrolling 2500-9999 reporting a computer related bachelor degree program being offered in 1976-77.

Alabama
Alabama State University
Auburn University at Montgomery

Arkansas
Arkansas State University
University of Arkansas at Little Rock

Colorado
Mesa College

Florida
Florida A&M
University of North Florida
University of West Florida

Georgia
Georgia Institute of Technology

Idaho
Idaho State University

Illinois
Chicago State University
Eastern Illinois University

Indiana
Indiana University at South Bend
Indiana University - Purdue University at Fort Wayne
Purdue University, Calumet Campus

Iowa
University of Northern Iowa

Kansas
Emporia State University
Fort Hayes State University
Pittsburg State University
Washburn University at Topeka

Kentucky
Morehead State University

Louisiana
Louisiana Tech University
Nicholls State University
Northeast Louisiana University
Southern University

Maryland
University of Baltimore

Michigan
Ferris State College
Northern Michigan University

Minnesota
Moorhead State University

Mississippi
Jackson State University
Mississippi University for Women
University of Mississippi

Missouri
Central Missouri State University
Northwest Missouri State University
Southeast Missouri State University
University of Missouri, Rolla

Montana
Montana State University
University of Montana

Nebraska
Kearney State College

New Jersey
Richard Stockton State College

New Mexico
Eastern New Mexico University

New York
SUNY College at Oswego

North Carolina
University of North Carolina at Wilmington
Western Carolina University

North Dakota
North Dakota State University
University of North Dakota

Oklahoma
Southwestern Oklahoma State University

Pennsylvania
Clarion State College
California State College
Cheyney State College
East Stroudsburg State College
Mansfield State College
Slippery Rock State College

Puerto Rico
University of Puerto Rico Mayaguez

South Dakota
University of South Dakota

Tennessee
Austin Peay State University
Tennessee Technological University

Texas
Angelo State University
Houston Clear Lake City
Texas Southern University
University of Texas at Dallas

Utah
Utah State University
Weber State College

Virginia
Christopher Newport College
James Madison University
Radford College
Virginia State College

Washington
Western Washington State College

West Virginia
West Liberty State College
West Virginia Institute of Technology

Wisconsin
University of Wisconsin Eau Claire
University of Wisconsin La Crosse
University of Wisconsin Superior

Wyoming
University of Wyoming

3) Private institutions enrolling less than 2500 reporting a computer related bachelor degree program being offered in 1976-77.

Alabama
Talladega College

California
Westmont College

Illinois
- Olivet Nazarene College
- Quincy College

Indiana
- Anderson College
- Depauw University
- Rose-Hulman Institute of Technology
- St. Joseph's College
- Taylor University

Iowa
- Central University of Iowa
- Graceland College
- Luther College
- St. Ambrose College

Kentucky
- Transylvania University

Maryland
- Washington College

Michigan
- Albion College
- Madonna College

Minnesota
- St. Mary's College

Mississippi
- William Carey College

Missouri
- Maryville College
- Westminster College

New Jersey
- Stevens Institute of Technology

New York
- New York Institute of Technology
- Vassar College

North Carolina
- Campbell College
- Catawba College

Ohio
- Denison University
- Mount Union College
- Muskingum College

Oklahoma
- Bethany Nazarene College

Pennsylvania
- Beaver College
- Geneva College
- Juniata College
- King's College
- St. Francis College
- Spring Garden College
- Waynesburg College
- Westminster College

Tennessee
- Freed-Hardeman College
- Lambuth College
- Steed College

Texas
- Austin College

Virginia
- Hampden-Sydney College
- Mary Baldwin College
- Washington and Lee University

West Virginia
- West Virginia Wesleyan College

4) Private institutions enrolling 2500-9999 reporting a computer related bachelor degree program being offered in 1976-77.

Arizona
- Harding College

Colorado
- University of Denver

Connecticut
- Quinnipiac College
- University of Hartford
- University of New Haven

Illinois
- Bradley University
- Illinois Institute of Technology
- Lewis University

Indiana
- Butler University
- University of Evansville

Iowa
- Drake University

Louisiana
- Loyola University in New Orleans
- Tulane University

Maryland
- Loyola College

Massachusetts
- Bentley College
- Massachusetts Institute of Technology
- Worcester Polytechnic Institute

Michigan
- Detroit College of Business

Minnesota
- College of St. Thomas

Missouri
- St. Louis University

New Jersey
- Farleigh Dickenson
- Monmouth College
- Rider College

New York
- Elmira College
- Iona College
- Mercy College
- Polytechnic Institute of New York
- Rensselaer Polytechnic Institute of New York

North Carolina
- Duke University

Ohio
- Antioch College
- Case Western Reserve University
- University of Dayton

Oklahoma
- University of Tulsa

Pennsylvania
- Drexel University
- LaSalle College
- Marywood College
- Robert Morris College
- University of Scranton

Puerto Rico
World University

Rhode Island
Brown University

South Carolina
Furman University

Tennessee
Vanderbilt University

Texas
St. Mary's University
Texas Christian University
Trinity University

Appendix B

Estimated Enrollments in Selected Computer Related Courses Reported in 1976-77 at Institutions Offering at Most the Bachelor Degree and at All Institutions

	At most bachelor enroll. below 2500	At most bachelor enroll. 2500-9999	All Insts.
Computer Appreciation	4,052	1443	40,443
Problem Solving Course in the Humanities	384	10	11,976
Introductory Course in Compiler Language like FORTRAN, PL/I, BASIC or ALGOL	14,148	6393	221,814
Introductory Course in a Language like COBOL	3,284	2714	47,464
Problem Solving Course in Engineering or Scientific Applications	1,791	324	30,622
Introductory Course in Prog.	3,459	386	96,088
Advanced Course in Programming	2,143	812	36,110
First Course in Computer Design	238	60	15,247
Advanced Course in Computer Design (Computer Architecture)	80	116	7,657
Programming Languages (Structures etc.)	107	557	17,183
Operating Systems (Systems Programming etc.)	312	207	17,682
Data Structures	407	102	16,836
Automata Theory	10	62	3,548
Simulation Applications (non-econometric)	188	150	4,266
Simulation Languages and/or Techniques	126	145	4,397
Artificial Intelligence	-	190	4,043
Hybrid Computing	-	-	607
Discrete Structures	165	38	7,122
Computer Organization	326	492	7,943
Switching Theory	174	19	4,569
Sequential Machines	124	12	1,367
Compiler Construction	152	43	6,757
Formal Language and Syntactic Analysis	34	12	3,266
Introduction to Computer Org.	91	119	11,305
Advanced Computer Organization	86	122	2,726
Information Storage and Retrieval	134	17	3,772
Computer Graphics	3	0	3,224
Computability	69	0	2,514
Introductory Numerical Analysis (or Methods)	778	786	19,179
Advanced Numerical Analysis	402	166	5,716
Introductory Operations Research	319	141	11,062
Advanced Operations Research	257	36	4,171
Data Communications	26	31	2,191
Process Control	72	143	14,381
Data Preparation	133	395	1,814
Unit Record	53	95	806
Introductory Course in Data Processing	2,917	6,845	77,421
Advanced Course in Data Processing	205	1,048	6,566
Problem Solving Course in Business Applications	269	59	30,297
Advanced Business Applications	131	69	6,341
Introductory Systems Analysis	372	802	9,495
Large Scale Information Processing Systems	26	66	964
File Organization	0	212	2,502
Data Base Management	14	121	5,402
Information Systems Design Introductory	124	1,026	7,134
Information Systems Design (Advanced)	203	178	3,374
Management Information Systems	402	691	8,491
Management of Computer Centers and/or Information Processing Departments	31	117	1,790
Computers and Accounting	107	69	2,683
Econometric Simulation Applications	383	0	1,259
Instructional Design	28	0	69
Instructional Systems	0	0	376
Computer Based Instruction	19	10	1,310
Other	1,183	3,716	52,150
Description Not Coded	3,469	0	40,538
Totals	44,102	32,084	948,606

USING PERSONAL COMPUTERS AS TERMINALS IN COMPUTER NETWORKS

Richard E. Horton
Associate Professor
Computer Engineering and
Computation Center
Iowa State University
Ames, Iowa 50011

ABSTRACT

A variety of services have been developed at Iowa State University to support the use of personal computers as terminals or as stand-alone devices with extended disc storage via connection to the ISU Datanet, a computer network. Users must, however, select from numerous options offered by computer manufacturers and those providing communications services before connecting personal computers to the Datanet.

The communication for the network is provided by a recirculating loop operating at an effective bit rate of 3.088×10^6 bits/sec. Two modes of communication are supported by the loop. Communication can be from dispersed stations to a central station or from one general purpose station to another.

The existing central facility consists of an interfaced PDP 11/34 with a UNIX operating system and 120×10^6 bytes of disk storage. Each centrally communicating station supports eight Commodore PETs which serve as stand-alone personal computers, as terminals on the UNIX system, or as personal computers with extended disk storage provided via the net. The central facility provides compilers, cross assemblers, language, and other support in addition to disk storage.

I. INTRODUCTION

A network of mini and microcomputers has been constructed and put into service (at the Iowa State Computation Center) to support the use of personal computers and microcomputers in experiment control and data acquisition with partial support from the National Science Foundation (1). The intent of the network is to enhance the effectiveness of researchers by providing support that would make personal and microcomputers simple and easy to use. The facility and educational material were designed to meet the needs of inexpert users; to this end facilities are centered around available personal computers and a well-established operating system.

The network facility does not explicitly address questions of accounting or security; the facility, including the personal computer and the communication facility, is simply provided on a monthly rental basis. Maintenance and support are provided in the normal way through the computation center, and security is assumed to be noncritical.

The communication facility was locally designed and built to meet system bandwidth and performance objectives with an intent of minimizing software requirements and operating costs.

II. THE SYSTEM AND SERVICES

A. The System

The system consists of a communication net, a PDP 11/34 central facility with 96K bytes of main memory and two 60M byte disc units, and more than two dozen Commodore PET personal computers. The PDP 11/34 uses the UNIX (a Bell Telephone Laboratory trademark) operating system to manage resources. In addition, a variety of single board computers are connected into the system.

The communication net, a recirculating network, is interfaced to the PDP 11/34 through DR 11C units. All PET connections to the communication net meet the RS 232C standards. In addition a communication link is being completed between the PDP 11/34 and the Itel AS-6 in the computation center to transfer large computational jobs.

Figure 1 depicts the facilities of the system; the parts of the recirculating communication network that are depicted are the master station (MS), the central stations (CS), the centrally communicating stations (CCS), and the general purpose stations (GPS).

B. The Services

A variety of services have been developed to support the use of personal computers and microcomputers. Key features center around the use of the Commodore PET (or other) personal computers; these can be utilized as net terminals on UNIX, as stand-alone devices, or as stand-alone devices that have extended disc storage via the net.

These features have been achieved by adding an IEEE 488 to RS 232C adapter to the PET and using about 1500 of the 7000 bytes of available storage for a "Netcom" program. (Netcom is locally developed communications software for the PET.) When the network and PDP 11/34 are used to provide extended disc storage, the "netcom" program automatically handles necessary protocol with UNIX and presents the facility to PET BASIC simply as another I/O device. In the terminal mode, only the keyboard, display, and I/O features of the personal computer are used.

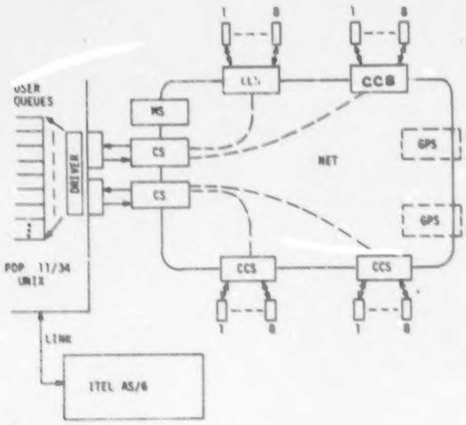


Fig. 1. The Datanet configuration.

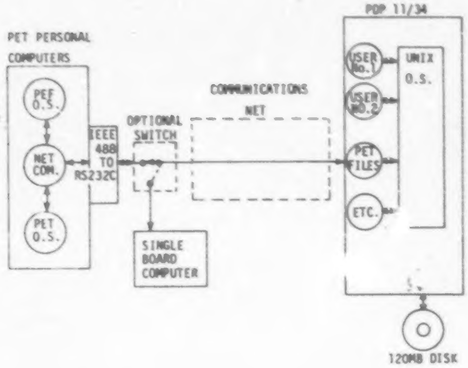


Fig. 2. Functional system diagram.

A number of other facilities are provided via the network. These include cross assemblers for a variety of microcomputers: the Intel 8080 and 8085 and the MOS Technology 6502. In addition, a C-like systems language for microcomputers is supported so that programs can be written in a high level language and object code can ultimately be loaded into a microcomputer. As noted in Figure 2 an optional switch is provided so that object code, when produced, can be simply downloaded into the microcomputer by switching the RS232C line connected to the personal computer. Figure 2 depicts key functional parts of the system hardware and software.

III. PET TERMINALS

A. Resident Features

Representation of characters in the main memory of PET personal computers is that of standard ASCII (American Standard Code for Information Interchange). Seven bits select one of 128 characters. The eighth bit is used to separate these standard characters from BASIC command words and special graphics characters for the PET screen (2).

Representation of characters in the screen memory (directly associated with the video display) is somewhat different. The 64 upper case alphabetic, numeric, and special printable characters may be uniquely selected with only 6 bits. A seventh bit selects between these 64 alphanumeric and 64 PET graphics characters. Finally, the eighth bit is used to select a regular or reverse video display for each character.

An additional variation allows selection of either the 64 graphics character set or a 64 character set composed of 26 lower case alphabetic and 38 graphics characters. This permits a display mode with both upper and lower case alphanumeric characters of the screen. Hardware selection for one of two character generator ROMs is done by POKEing a special constant to an internal I/O (Input/Output) port.

Conversion between screen memory code and ASCII is handled by the PET operating system or monitor. Of the potential 32 ASCII control characters generated by the keyboard, only CR (carriage return) is implemented. The RETURN key produces CR.

Another consideration is that only parallel I/O is implemented on the PET through an IEEE-488 (Institute of Electrical and Electronics Engineers Standard #488) (3). Most computers expect to attach to terminals with serial I/O. However, the PET monitor is designed to exchange parallel data with peripherals through the IEEE-488 interface. This is very convenient from a users viewpoint provided the peripheral has a compatible IEEE-448 interface as well.

B. Added Features

Problems arising from resident features of the PET used as a terminal include 1) I/O is parallel, not serial, 2) No ASCII control character keys exist with the exception of CR, and 3) the eighth bit is not associated with a standard parity option.

One solution to these problems involved a combination of additional hardware and software. A PET Serial Adapter was designed as a hardware attachment to the IEEE-488 I/O interface to convert parallel data to serial data for transmission and serial data to parallel data for reception (4). All handshaking functions of the IEEE-448 data exchange protocol are implemented in hardware. Additionally, strapping options exist for device address number, baud rate and the desired number of stop bits in the standard asynchronous serial data format. The most significant bit is not used for parity, rather to indicate the status of a byte to be exchanged between the PET and the Serial Adapter. Full duplex communication is necessary and possible over an RS-232C voltage interface or an active or passive 20 milliampere current loop interface (5).

A machine language program, Netcom, handles additional requirements with software. The keyboard and Serial Adapter are alternately scanned for input, and output is routed to the Serial Adapter or PET screen memory. A "shift-$" graphics character is replaced with the ASCII control character EOT, End Of Transmission, whenever it is detected by a scan of the keyboard. Upper and lower case alphanumeric case and normal video modes are selected. Monitor address pointers are adjusted to set a new address fence which inhibits user programs and data from encroaching into the memory space where Netcom is located.

IV. EXTENDED DISK STORAGE VIA THE DATANET

A. Additional Netcom Features

A large portion of the Netcom software is concerned with utilizing extended disk storage facilities on the Datanet by accessing them through the Serial Adapter.

Part of the PET monitor is a "command analyzer" concerned with BASIC commands issued from the keyboard or through normal program execution. BASIC I/O commands dealing with the physical exchange of programs or data between the PET and a peripheral have a device number associated with them. Characters are exchanged between the PET and one of two cassette tape recorders, devices 1 and 2, through separate 192 byte buffer areas in the PET RAM memory. Netcom adds two new special external device numbers (101 and 102) to the list of permitted device numbers (4 to 40). Devices 101 and 102 are physically associated with the PDP 11/34 central facility disk units. Physical access to files on these units is through the Serial Adapter associated with the RS-232C serial 1200 baud terminal connection to the Datanet. Internal cassette buffer areas for drives 1 and 2 are shared by these new devices 101 and 102.

The BASIC command SYS(n) is used to transfer program control to a user program at location n. Netcom adds two additional variations to this command which are detected through the command analyzer in the PET monitor.

SYS without any argument directs program control to Netcom which puts the PET into terminal mode. Return from this mode is accomplished with the STOP key.

SYS "character-string" is another non-standard variation permitted by Netcom which causes the PET to go into terminal mode and issue message strings to the host computer, in this case the PDP 11/34 central facility. Automatic return from terminal mode occurs upon receipt of an EOF (End Of File) from the host.

Implied with the transmission and reception of program and data files between the PET operating as a stand-alone computer with extended disk storage and the host computer with this extended storage is a message protocol running in the PET as well as in the host. Netcom interprets various messages which may be received from the host and formats messages for serial transmission to the host. Another program, Petfiles, was written to run on the PDP 11/34 to compliment the message protocol features of Netcom.

Finally, all BASIC programs and data are stored on the extended disk in ASCII-hexadecimal character load string format. Although this cuts the effective data rate by at least a factor of 2, several advantages exist. This load string format is identical to that for object code produced by cross assemblers and compilers available to Datanet users. This means that Netcom need not be concerned with whether the file being down-loaded from extended file storage is a BASIC program, data, or object code. The format is identical and can be handled in a uniform manner.

Combined with UNIX file management and text editor software available to Datanet users, the PET personal computer becomes a terminal attached to a very powerful microprocessor/microcomputer development system with the added capability of operating as a stand-alone computer with extended disk storage.

B. Summary of Commands Supported

BASIC file statements which may be associated with extended disk storage through use of device numbers 101 and 102 include LOAD, SAVE, OPEN, CLOSE, PRINT#, INPUT#, and GET#.

The BASIC command SYS(n) is augmented to include SYS and SYS "character-string".

Representative UNIX file commands which can be issued in terminal mode include

ls	list contents of directory
chdir	change working directory
cat	concentrate and print
mail	send mail to designated user
mv	move or rename a file
rm	remove a file
ed	invoke the text editor

V. REFERENCES

1. Pohm, A. V., Davis, J. A., Christiansen, S. R., Bridges, G. D., and Horton, R. E., "A Local Network of Mini and Microcomputers for Experiment Control", 4th Conference on Local Computer Networks, Minneapolis, MN, August, 1979.

2. PET Reference Manual, Commodore Business Machines, Santa Clara, CA, 1979.

3. IEEE STD 488-1975(78), Digital Interface for Programmable Instrumentation, Institute of Electrical and Electronics Engineers, New York, NY, 1975(78).

4. PET Serial Adapter, Computer Associates, LTD., Ames, IA, 1979.

5. McNamara, John E., Technical Aspects of Data Communications, Digital Equipment Corporation, Maynard, MA, 1977.

Can the User of a Large Time Sharing System be Satisfied with a Microcomputer?

J. F. Huntington
Miami University
Oxford, Ohio

Within the past few years, microcomputer systems have become available and are competing with other computer systems. As microcomputer manufacturers provide modular add-ons the distinctions between microcomputers, minicomputers and full scale computers become blurred. Many microcomputer systems provide 64K or more of RAM/ROM memory which places them in the large computer category of the past decade. With disc operating systems (DOS) which provide 80-100+K increments of up to .5 x 10^6 K or more of fast, I/O storage, microcomputer systems lose the distinction of limited size. The large time sharing system (LTSS) user would be largely unaware of differences encountered when using a microcomputer terminal except several important system's operating requirements and characteristics.

An important consideration for judging microcomputer systems is the phenomenal development of I/O devices by microcomputer manufacturers and peripheral suppliers. The time lag for development and marketing of these devices is considerably shorter than that currently required for LTSS devices. They are offered at lower prices and probably offer lower reliability than their LTSS counterparts which are often overdesigned to permit high reliability in business oriented activities. In fact, many of the microcomputer I/O devices are primarily oriented toward the hobby market and may or may not be acceptable to the intensive computer user. The existence of a large market and rapid development/marketing procedures have resulted in an extraordinary array of interesting I/O devices for microcomputer systems. Some of these devices permit audio I/O, opscan/marksense input, light pen input, joy stick input, printing output, time sharing I/O, servo-switching, time/temperature input, color VDT output, and, in fact an almost limitless array of electro-mechanical devices that can generate or receive encodable signals that are microcomputer compatible. Another issue to consider is that microcomputer peripherals are generally end-user oriented in terms of installation, maintenance and operation while LTSS devices often require the specialized attention of a trained system's operator. The essential differences between LTSS and microcomputer I/O devices are derived from different marketing philosophies about economics, marketing, reliability and considerations of user characteristics. Microcomputer I/O devices offer a rich variety of potential uses previously unavailable to many LTSS users who don't have the services of computer hardware technologists readily available.

The essential development which provides real competition between microcomputers and LTSS systems is that of the DOS microcomputer (MDOS). Response times for MDOS are quite similar to those encountered with LTSS systems with multiple users. When heavy CPU demands are made by several users of LTSS, response time can degrade to several seconds, a condition which is not common to microcomputer systems since only one user is accessing a CPU of nearly equivalent size. MDOS systems are quite good and replicate LTSS operations with good results. Programs and files can be saved by name or number, chaining is possible and LOAD/WRITE times are quite acceptable but generally slower than most LTSS. Multiple disc drives permit storage of long programs and files though LOAD and search times are increased. On the whole, MDOS systems are quite competitive with the file manipulations provided by the LTSS. Disadvantages of MDOS system's operations will be discussed later.

Perhaps the greatest consideration undertaken when potential computer users contemplate purchase of computing resources is cost/performance. Purchase of LTSS computer time and a terminal may be less costly than acquiring an MDOS. The significant problems that many potential users face are (1) access and (2) phone line charges. These factors may prohibit LTSS use to many and microcomputer acquisition may be the only viable alternative. Generally, when considering original purchase price and maintenance the costs of LTSS and MDOS are nearly equivalent when phone line charges are not excessive. The greatest cost factor that is known at this time which might affect the MDOS user adversely is linked to operations. Equipment reliability and longevity factors for microcomputer systems remain largely undetermined.

The MDOS required considerably more user/operator activities than the LTSS. Part of the LTSS service includes the operator functions of loading discs or tapes, initializing the system, providing maintenance and informing the user of special system status checks. All of these functions, in their most intimate detail, are required of the MDOS user/operator. Some of the concerns about MDOS requirements for smooth operation are:

regulated power supply for MDOS reliability

MDOS temperature/humidity operating environment

disc head cleaning/demagnetizing

data backup system

environmentally safe disc storage

DOS debugging capability

disc loading/unloading

maintaining sufficient CPU capacity

disc library management

equipment installation

maintenance record keeping

system's operations referral potential

program backup system

system maintenance

consultation about system characteristics

scheduling of system use

maintaining system reference library

integrating system updates from manufacturer

communicating to users

etc.

Design limitations have been cited in microcomputing literature by users of various systems. Some of these limitations have been rectified by the respective manufacturers. During the past few years it has become evident that microcomputer manufacturers must rapidly rectify malfunctions in their systems if they are to remain competitive. Their reactions to these problems appear to be quite responsive. This may be explained in part by the demands placed on the manufacturer by hundreds of thousands of users through direct contact rather than through field representatives. The outlook for manufacturer response time regarding design updates and patches is promising. Some of the limitations which have been reported are:

faulty disc loads

keyset bounce

end of file problems

faulty disc writes

system "lock-up"

lost file labels

poor electrical connections

poor documentation

nonstandardized I/O connections

keyset configuration

The procedures required for using the LTSS are less involved than those required for the MDOS since an operator performs some of them at the LTSS computer site. A comparison of the steps required for a job follows.

LTSS

1. Terminal-on
2. Initiate coupling if required
3. Enter job
4. Close job
5. Terminal-off
6. Secure terminal site

MDOS

1. Uncover system
2. Power-up (system, printer, etc.)
3. Locate disc from library
4. Load disc
5. Enter job
6. Manage disc I/O
7. Close job
8. File disc(s) in library
9. Power-off
10. Record use time, materials consumed
11. Secure equipment

The tasks required uniquely by MDOS are not excessive and provide two system's characteristics which might be more advantageous than disadvantageous. First, disc management is controlled by the user rather than by a remote operator while LTSS systems often permit more space for program and file storage. File manipulations are easier on the LTSS because loading and unloading are not user required. Chaining is also assured because no interrupts occur. The issue of user versus operator/system control of disc loading/unloading is the critical decision which involves user autonomy. The other issue involves file security. LTSS systems may promise security but it still depends on an effective management system with zero breakdowns. Security for the LTSS system is essentially person dependent and risk taking must be evaluated on the basis of how much integrity is displayed and predicted for the system. MDOS shifts the security from off-site to strictly on-site considerations. If the system can be secured in locked space, violation can only occur by instrusion or personnel leaks. The controls for the MDOS are local procedures which may be advantageous for some users.

The MDOS system appears to be competitive with LTSS systems. Users contemplating the purchase of an MDOS system should carefully consider the talents of local personnel. MDOS systems require operator skills which LTSS systems provide. The advantages and disadvantages of MDOS should be carefully evaluated before purchasing. MDOS purchases are more substantial than terminal purchases and users should weigh the implications of hardware acquisition in terms of obsolescence, maintenance and upgrading. MDOS has much to offer certain users. These systems considerably broaden the population of potential computer users.

101

MICROCOMPUTERS
IN THE ELEMENTARY SCHOOL:
WHAT EXPERIENCE HAS TAUGHT US SO FAR

Willis Jokela
Minnesota Educational Computing
Consortium (MECC) Saint Paul, MN

Introduction

The Minnesota Educational Computing Consortium (MECC) has been supporting instructional computing in Minnesota for over five years through a large centralized timeshare system. As a result of a task force study, MECC is now also supporting the use of microcomputers in Minnesota's educational institutions. Through a bid process in October 1978, the Apple II microcomputer was specifically selected for support and software development. A little more than a year later, over 850 Apple microcomputers have been purchased by educational institutions in Minnesota.

To begin with MECC converted and enhanced about 80 programs from the large timeshare computer to the Apple microcomputer. These programs were formatted for screen output, with graphics, color, and tones added when appropriate. After the initial conversion MECC concentrated on total packages for curriculums and programs that were requested by schools. It was hoped that schools purchasing the microcomputer would do more converting of other timeshare programs needed.

The Apple microcomputer was first introduced to schools through regional workshops. Follow-up workshops were then held at schools showing an interest in microcomputing and how the computer could help in the classroom. Some schools went ahead and purchased the Apple, while others waited until teachers expressed an interest in using it.

This paper will present some things the author has learned from viewing the use of microcomputers in elementary schools in Minnesota concerning the caretaking and use of equipment.

Organizing For Teacher Use Of Microcomputer

Schools generally have one person in each building in charge of the microcomputer and one district person to coordinate getting materials and writing programs for the teacher who does not have the time or the knowledge to write his own programs. The librarian has usually been the person chosen as the building coordinator. Some librarians were unwilling or unable to take the responsibility, so a teacher wishing to use the microcomputer became the coordinator. However, if the building coordinator is someone other than the librarian, teachers will tend to shy away from using the microcomputer. The main reason for this is that the teacher-coordinator has only time to prepare for his own class and can make little time for demonstrations to other teachers.

When there are no demonstrations of the microcomputer, teachers tend to keep their fear of the computer longer and do not take time to implement the computer in the classroom. Schools have found that older students can help promote computing by having as a class assignment the introduction of the computer to a teacher unfamiliar with it. The student shows the teacher how to use the computer and what programs are available.

The librarian will usually introduce the students to the computer and how it works when they come in during their normal library times. A district person should be given enough time to set up and give literacy classes to students and large workshops to teachers that explain everything from the handheld calculator, to toys that use a microprocessor, to microcomputers, to timeshare computers. The teacher's fear of the computer will begin to disappear with workshops and when the students begin asking when they can use the computer.

In schools where one teacher has the microcomputer in the classroom all of the time, other teachers have asked why they cannot have one for their classroom. This is one reason why the computer should be used as a resource unit located where teachers can sign it out when needed. There are usually very few problems the first year with conflicts between teachers until they lose their fear of the computer and start working it into their curriculum. After the first year an evaluation can be made to see how many teachers are really planning to use the computer and if more microcomputers should be ordered for the next year. A school will probably need one micro per every one hundred to one hundred and fifty students within the next five years.

Preparing The Microcomputer For Student Use

The use of equipment must be simplified. An effective location is an AV type cart. A larger top should be built so there is room for the microcomputer, disk drive, and a place to write. A shelf above the micro is needed for the monitor or television. The shelf should be at a distance so that the microcomputer's top cannot be opened by students. The diskettes should be kept in a binder, one to a page, and letter-coded to avoid confusion, with one page explaining what is on each diskette. The cart should have a long extension cord with a multi-plug end so that all parts of the computer can be connected to it at all times and the student need only plug in the single cord.

Another good cart is a metal cabinet that can be closed and locked. It should have a shelf in it that pulls out for the microcomputer and disk drive, with the television placed on top of the cabinet. These short carts are just the right height for typing, yet low enough for the elementary student.

What Elementary Students Do With Microcomputers

Students in grades five and six can understand what a computer can do and can identify its parts. They understand that it is sensitive and the keys do not have to be pounded like an old typewriter or piano to get it to work. In fact, some seem afraid they may break the computer. The teacher introduces the microcomputer by taking the class through each program a few times and then giving the students the key words and instructions needed. The directions for the student can be very short because they will memorize what is needed and will skip any other instructions given.

Many different programs are used by students at this level. The states, their capitols and locations, can be introduced in the classroom through a computerized drill program that produces maps for the student. Programs that use logic are very popular, such as one in which the student tries to find an object hidden within a grid. With a grid the computer can help a student learn about the coordinate system, distance on the grid, and about directions such as north and south.

The concept of a food chain can be learned through a simulation in which the computer simulates the interaction of fish in a pond. Music teachers have found the microcomputer to be a great assistant in making music that can be very quickly repeated. Students have been able to run the spelling programs in which they enter the words in the computer and receive help from the computer with learning the correct spelling. They even learn how to use the computer to correct errors. Math drill and practice on the computer has been used by teachers for contests between groups of students. The computer randomly generates the problems and keeps track of the scores.

Third and fourth graders become very excited about getting to use the computer; one problem for them may be in getting a 'feel' for the keyboard. Concepts of north and south, left and right are quickly learned and students are able to run programs on map reading and working with a grid. They also use programs that drill on adding and multiplying of one table at a time such as by threes and fours. Computer programs should have much movement in a picture and other reinforcement to hold the students' interest. Directions are of little or no help so that teachers must take students through a program many times, but eventually, students learn to run the computer on their own.

Students in grades as early as kindergarten can use the computer, but mostly in a gaming situation. They can do limited kinds of map reading if they know north and south and left and right, or if the program uses 'one' as up, 'two' as down, etc. Programs students learn the most from have to do with the alphabet, such as, What comes before or after the letter "C"?, or, What comes between the letters "E" and What comes between the letters "E" and "G"?

Programs currently needed in the elementary area and being worked on at MECC cover topics such as fractions, geometry, prefixes, and spelling with the help of a tape recorder.

Conclusion

It is clear that elementary students enjoy using computers and can use them effectively. It is not long before they willingly stay after school to use microcomputers, causing parents to wonder why their children have suddenly forgotten when to catch the bus home! However, an elementary school is a different environment for a microcomputer than a high school. Questions such as who has charge of the equipment, how equipment should be set up, and what materials should be available must be answered in unique ways when providing microcomputers in an elementary school.

MANAGEMENT INFORMATION SYSTEMS - PROGRAMMING FOR INDIVIDUALS WITH EXCEPTIONAL EDUCATIONAL NEEDS

Wayne A. Jones
Onondaga-Madison BOCES
Syracuse, New York

Marilyn Seemann-Jones
Management & Motivational Systems
Syracuse, New York

The purpose of this paper is to provide an overview of the differences between the needs of general education and exceptional education in electronic data processing and management systems. The "ideal" components of a management information system for individuals with exceptional educational needs will be outlined. The outline is derived from discussions of computer utilization, decision making, program evaluation, applied research and the differences between regular education and exceptional education.

Utilization in Exceptional Education

As the data requirements for the substantiation of program accountability increase, educators in exceptional education not able to incorporate computers into management technology will be left with an overwhelming task. In the fact of the increasing rate of data demands, educators do not appear to be increasing computer utilization at a corresponding rate (Mathews & Sullivan, 1979)

The use of computer technology in the development of reports and management of educational systems provides four major advantages. A computer reduces the amount of time required in the development of extensive and complicated reports. High speed digital computers can complete calculations and provide hard copies of lengthy reports more quickly and accurately than possible through clerical (manual) systems. The capacity of the computer managed system to handle large data files provides the educator with the flexibility of collecting, storing and retrieving information with accuracy and efficiency. In situations where the educator is involved in the development and implementation of complicated management systems, the ability of the computer to solve simultaneous calculations necessary in the operation of such systems cannot be duplicated manually. Finally, computer simulations and analogues provide the educator with opportunities to investigate potential problems and operations of complicated management systems under varing conditions without suffering financial loss or political embarrassment.

For educators providing services to handicapped students, the primary impetus for incorporating computers into their management technology is the growing body of litigation and legislation mandating due process and equal protection for handicapped individuals. The need to document due process activities and to provide information to funding agencies should increase the computer utilization rate of simple aggregate or statistical reports; however, some educators are beginning to deal with the more complex task of providing information to decision makers at the classroom and student levels as well as the administrative levels. General or regular education has utilized computer reports in school finance, payroll, food services, text book inventory, equipment inventory, district budgets, standarized testing, report cards and class scheduling.

The utility of these reports has been predicated on their usage in large population centers or by consortia in order to reduce unit costs. As state and federal litigation and legislation began to increase the numbers of identified and served handicapped individuals in education, Electronic Data Processing applications became more commonplace in the education of individuals with exceptional educational needs (special education). As in regular education usage, however, administrators of programs in exceptional education have oriented their systems toward the administrative level more than classroom level applications. Consequently, many of the student, personnel and financial file structures used in regular education can be applied directly (or with minor modifications) to administrative needs in exceptional education.

In a recent study conducted by the Management Analysis Center, Inc. in Washington D. C. (1979), five state educational agencies and five local educational agencies utilizing computer based management information systems were reported. The "model" systems represented a, "...range from small to large and to relatively simple to complex." Of the ten systems, one (Great Falls Local Education Agency) was oriented to the monitoring of goals for individual students.

Onondaga-Madison Board of Cooperative Educational Services in Syracuse, New York operates a data based management information system, in exceptional education programs, designed to monitor student performance and facilitate appropriate decisions at the classroom and district levels. The primary report generated by the system is an individualized educational program which maintains a longitudinal record of student performance and instruction provided on an unlimited number of behaviorially stated objectives. As the report results from the merging of three files (student, teacher, objectives) the content of the reports generated is limited only by the data elements. Since personnel, teaching materials, and hours of instruction are data elements, and since the file structures permit the inclusion of cost information, cost-benefit as well as program analyses is possible.

The Onondaga-Madison BOCES System has three exemplary aspects. The system collects longitudinal information and maintains the file so that several years of data may be available in one individual's educational program. The system has reduced the amount of time required by teachers and clerical staff to complete and maintain information on student programs. For a relatively small number of moderately to severely handicapped students, the unit operation cost is expected to be less that $7.00 per student during the system's second year of operation. Since the system was developed and implemented by existing BOCES staff, state or federal funds were not required.

Administrative and Management Decisions

For most educational agencies, student enrollment in exceptional education programs represent a source of revenue as well as expenditure. Computerized management information systems may be used to service these enrollment processes. Applications for enrollment-generated revenue are from various local, state and federal agencies. Although each of these agencies require some mutually exclusive data, most of the students' demographic information is amenable to collection and maintenance on a common system. In addition, long term planning, based upon enrollment information is often related to statutorally required planning (one year or multi-year educational plans), annual budgets, tax levies, bonding issues, numbers and types of programs, and vendor selections (local, cooperative, residential or private). Short term planning, based upon revenue information (often occurring on a week to week basis) is typically related to transportation, staffing, consultation, material and equipment. A computerized management information system is therefore a viable alternative to manual systems because of the more efficient, and accurate service available in enrollment maintenance, financial accounting and long and short term planning.

In the areas of personnel management at least four important decisions may be serviced by a management information system: personnel selection, termination, in-service training and assignment. Personnel selection and termination decisions most often are affected by available revenue and student needs assuming that personnel tenure areas are a function of student programming rather than vice versa). Personnel in-service training decisions are most often the function of student needs, inadequate pre-service training and current developments in educational practices and legislation. Personnel evaluations can provide information upon which classroom decisions and management decisions may be made; therefore the lowest common denominator of personnel selection and assignment becomes student performance on behaviorally stated objectives with measurable criteria. If an adequate management information system is operational, personnel assignment and in-service training may be based on individual student needs.

Based on current categorical models, the most spurious of administrative decisions, student placement, is often made with the least amount of valid information. Students are generally placed into exceptional education programs based on the degree of deviance they exhibit on measurement instruments which are standardized on "normal populations." The use of management information systems in special education offers useful alternatives to the placement of students into categories of disabilities in order to provide funding and service (a common practice). Most obviously a longitudinal record of the actual human and other resources required to achieve specific educational outcomes provides the capability of defining unique program and resource requirements in terms of individual educational needs as opposed to simply putting the student in the special education class.

Classroom Decision Information

Three major classroom decisions which may be supported by a management information system (with a data base in student performance) are the selection of appropriate student responses, the selection of appropriate antecedent stimuli and the selection of appropriate consequences. The selection of the appropriate student response may be derived from a data base of available objectives arranged in a developmental or prerequisite order. Depending on the student's strengths or completion of prerequisites, as recorded in the student data base, the teacher may make more enlightened selections of student objectives.

The selection of antecedent stimuli (cues or instructions for students) from numerous alternatives may be more effective if their selection is based upon a student data base. Not only is the teacher's selection of the input-output modality (e.g., look, say) important, the teacher must decide upon appropriate stimulus fading procedures, modeling, shadowing, prompting, magnitude, duration, and other characteristics of the antecedents to a desired response. Obviously, the number and possible combinations of alternative antecedents is difficult to deal with without support from a management information system.

Consequences are as important as responses and antecedents but more difficult to collect and maintain. The selection of consequences of the student's response are seldom found as part of a data base in a management information system but, given a highly sophisticated system, they may be included.

Program Evaluation Information

Program evaluation should include staff evaluation and is defined here as the development and collection of information upon which program decisions may be based. The components of a program evaluation which may be served by a management information system are needs assessment, implementation evaluation, process evaluation and product evaluation (Klein, 1971).

Needs assessment or context evaluation requires that the activities within a program be articulated and described using methods similar to those inherent in the design of management information systems. Having established the goals and objectives of a program, a management information system can provide information on the discrepancies between a program's current level of functioning and its goals and objectives.

Once the time lines for the activities designed to reach a program's goals and objectives have been established, a management information system may generate individual or aggregate data at various points along the time lines as a method of evaluating the programs implementation. Process evaluation may be conducted using a system which permits the reporting of rates of flow of students through objectives or programs.

Product evaluation is the single most important use of the management information system, either in the individual educational program sense or the disability (categorical) program sense. The most important measurement of student performance or program performance is the rate of performance or acquisiton of objectives. When the student's rate of performance is related (through the management information system) to classroom decisions, to management decisions and to statutory requirements (i.e., categories, definitions, testing requirements), more enlightened decisions are possible by both managers and teachers.

Applied Research

Depending on the soft-ware used to develop the management information system, a wide variety of research techniques are possible. The techniques and statistical analysis will not be discussed here. Some of the questions, however, which may be answered using various research techniques and the management information system follow:

1. Are Disability categories appropriate for differentiation of treatment and/or cost?
2. Are various rates of performance on specific response classes (objectives) predictable given certain characteristics of students?
3. What programming or sequencing of programs are most cost effective?
4. What pre-service training (as reflected in academic degrees, certification, etc.) is most highly related to increased rates of performance by specific students on specific objectives?
5. What sequence of responses are most often successful for students with common characteristics?
6. What antecedents and materials are associated with higher rates of learning for specific students?
7. Which responses or objectives are most often and least often used?

Differences Between Regular and Exceptional Educational Information Needs

Individual programming for handicapped students is required by state and federal law. While scheduling by curriculum groups (classes or sections) is generally accepted as sufficient in regular education, exceptional education requires individual education programs which seldom permit academically homogeneous groupings. When groupings occur in regular education, the level (grade or age) of the student's functioning is the primary consideration. Although exceptional education may consider the level of the student's functioning, the student's rate of learning is of paramount importance. For example, the grouping of three students, all reading at a third grade level may be inappropriate if one of the students is four years older than the other two. Additionally, if one child of the group is learning at a rate of one grade per year while the other two are learning at a rate of one and a half grades per year, the grouping is inappropriate.

Finally, the product of exceptional education systems is the acquisition of specific objectives, while the product of general education is often the passing of Regents' examinations, high scores on an S.A.T. or the acquisition of a diploma. Management information systems designed for servicing decisions in exceptional education may, therefore, be useful in regular education; however regular education systems which do not emphasize individualized programming cannot serve exceptional education decisions adequately.

Comparable problems exist within the area of testing. While a regular education system might use norm referenced achievement tests as part of the data base, exceptional education programs often obviate the utilization of norm referenced tests. Also, since the exceptional education student's objectives must be stated in measurable terms (with criteria) the maintenance of norm referenced test data in a management information system is usually irrelevant (except as required by state or federal regulations). Pre-post testing using norm referenced tests may appear to provide an objective evaluation of a student's progress; however no rational argument can support the utilization for programming of samples (tests) of student learning when longitudinal data on all objectives within the program are available in the data base of the management information system.

Often, the rate of learning of the more severely impaired students is so low that the usual comparisons of levels or grades per year are impossible or provide no useful information. For example, while the rate of learning per year (by grade or level) may be useful in programming for reading activities, the rate of acquisition of the toilet training chain of behavior or the rate of acquisition of certain syntactical features by an adolescent may extend over several years. Therefore, the data base for the severely impaired does not fit into the usual grading or leveling conceptual frame of reference.

The data base for severely impaired students, if it is to be used to express rate of learning, must be longitudinal in nature and facilitate comparisons of the rates of learning of various responses (objectives) by one student. Comparisons of variations in a student's rate of learning of a response class or of a specific response can be used to assist the teacher in developing more efficient individual programs. Further, several variations in the physical conditions of severely impaired students may be detected through the observation of variations in rates of learning. These variations then become diagnostic data which are often helpful in prescribing medication and other physiological treatments. Due to the lower rates of learning and often concomitant physical changes in severely impaired students, data must be maintained over much longer periods than is usually considered necessary for students in general education. In order for the data base of a management information system to service the programming needs of the severely impaired (which are often characterized by regressions and plateaus over periods of several years) the data base cannot be erased or "dumped" for storage on a yearly basis.

Guidelines for Systems Designs

Joseph Orlicky provides guidelines for the systems' designer in the planning of a computer support system (1969). They are:

1. Make certain that upstream requirements are met. That is, higher echilon managers (e.g., boards of education) understand and approve computer utilization.
2. Develop clear and explicit program specifications.
3. Automate the function itself, not just its procedures. That is, the entire management structure should be oriented through a systems design to computer usage rather than many individual operations within the management structure.
4. Integrate but avoid complexity. That is, in the development of the single system inappropriate combinations requiring complicated linkages should be avoided.
5. Design a system that is modular. That is, each major component or module of the computer operation should be able to function separately yet be linked together in accordance with the logic of the mutual input-output relationship.
6. Establish what the proper level of automation should be. A fully automated system may be neither necessary nor appropriate.
7. Avoid need for informal systems. The automated systems should be able to operate without ancillary clerical or computer support.
8. Provide for auxiliary system functions. That is, in addition to primary systems' functions, the program should provide the necessary flexibility for the development of additional reports or functions.
9. Make systems' organization independent. That is, the designer should guard against systems obsolescence as a result of minor changes in the organizational structure or management system's design.
10. Anticipate the systems' effect on its environment.
11. Provide for the systems' growth.
12. Keep service to user uppermost in mind.

These guidelines are provided with the intent that they may be useful in consultation situations with computer management specialists. In addition to the components suggested by Orlicky the systems designed for exceptional educational needs should include the following:

1. Data-Based System - The system should be data-based in nature. At least four files (student, objective, staff, health) should be structured to permit interactions which would facilitate the most frequently required reports.
2. Intensity of Instruction - Either through some combination of data maintained in the student and objectives files or contained in one

file, the amount of time actually spent on an objective should be included. This may be done by reporting the actual time an individual teacher spends on an objective on a daily basis or the approximate time the teacher spends on an objective during a selected reporting period. In any case, the time must be related to a specific teacher on a specific objective.

3. Common Taxonomy - Users of the system must be provided with a "Thesaurus" of terms used in the systems operation. Standardization of the terminology used to operate the system is critical for both parent and staff training as well as validity in data interpretation.

4. Continual File Update - Since the reporting date for each student may be variable and since staff and objectives are constantly changing, the systems' operations cannot be constrained by limiting the frequency with which the files may be maintained (updated).

5. Common Coding - Data elements should be titled and coded using those titles and codes most often required by agencies requiring reports, such as the State Education Department. Although such reporting agencies may not always accept the systems-computer generated reports, common coding and titling will permit more efficient clerical manipulation of the data contained in the systems' reports.

6. Health Records - The health file or physical condition file for students with severe impairments should include information describing the students' "normal" condition. For example, many severely impaired students have normally high or normally low body temperatures or body weight. Fluctuations in the severely impaired student's temperature and weight toward population norms may indicate physical distress or disease. Longitudinal files on these fluctuations as well as fluctuations in hearing, vision, medication, combination of medication, seizures, aberrant behavior and diet may be directly related to the student's rate of learning as well as survival.

7. Longitudinal Data - Due to the lower rates of learning in severely handicapped students, the file structures must permit the maintenance and retrieval of information over a period of years. Many of the conditions related to appropriate programming for handicapped students cannot be observed over a one or two year period. Not only the rate and sequence at which a student acquires objectives but also the occurrence of regression or deterioration over a period of several years must be considered in programming for the individual student. The "dumping" of data onto storage tapes at the end of each year does not meet this criterion.

8. Student Centered - Most regular education systems are centered on the class, section, or teacher. Due to the individualized nature of programming for students with exceptional educational needs, the student must be seen as the center of the file structures with the teacher treated as a resource assigned to the student. This is especially important in most situations where two or more staff (the teacher, the speech pathologist, the psychologist, the occupational therapist) provide services on one objective for a student.

9. Input Document Limitation - When possible, "turn around" documents rather than separate input and output forms should be used. Limiting the number of forms requiring teacher time is critical to the success of a system.

10. Limited Clerical Manipulations - The number of errors occurring in transactions will increase as the complexity and number of transactions increase. If possible, the clerical manipulations should be restricted to the smallest possible number of editing, verifying, and transcribing operations.

11. Avoid Abbreviations - Because of the number of individuals and agencies interacting with any one student with exceptional educational needs, abbreviations should be avoided. If abbreviations or codes are necessary, explanations or supplemental information must be provided. The supplemental information is also a viable alternative to reducing the number of data elements and therefore reducing costs.

12. Provide IEP's - Since individual educational programs, are the most costly and widely referred to reports developed in exceptional education, the system must be designed to generate all or the major statutorally required components of the IEP.

13. Statistical Analysis - The system should be compatible with commonly used statistical soft-ware packages, such as SPSS, in order to provide

the systems' data predictive validity.

14. On-Line Capability - If an on-line interactive environment is not immediately necessary, the system should provide for the capabilities of upgrading to that environment if necessary in the future. On-line, interactive capabilities are costly and not always necessary. Experience has shown at the Onondage-Madison BOCES that a one-to-two-week turn-around time or batch processing is sufficient. Most individualized educational programs contain a sufficient number of well sequenced objectives to permit the instructional staff to complete an objective, report and proceed to the next objective while waiting for an updated IEP.

15. Permit Manual Operation - The format in which the reports are displayed as well as the file maintenance should be compatible with an effective manual operation. This is important in that it will permit smaller educational agencies to approximate a systematic management of their information and may encourage partial if not complete implementation of the system. Manual capabilities also permit the diagnosis of problems within the computerized system and facilitate in-service training.

Summary

Two major differences appear to be present in the development and operation of computer managed information systems for the non-handicapped and the handicapped. First the programs for the non-handicapped (regular education) tend to be oriented toward administrative decision making. The files are structured based on class or section assignment and student assessment is usually through norm referenced procedures. Student performance is often reported on a stanine or quartile scaling and is, therefore, related to the performance of peers. Programs for the handicapped (exceptional education) on the other hand, must be related to the individual student. Secondly the file for the non-handicapped students may be structured on a large population using a relatively fixed curriculum on a year to year basis, while the file for the handicapped students must be structured on a small population (with a larger number of data elements), using a changing and flexible curriculum with data maintained over a number of years.

Due to the increased number of individuals receiving services since the advent of mandatory legislation, computerized management information systems have become viable alternatives in support of exceptional education. Additionally, the management of a larger number of data elements over longer periods of time, has rendered manually operated systems inefficient and ineffective. The computerized management information systems for individuals with exceptional educational needs can be designed to service both administrative and classroom decision-making efficiently and effectively.

References

Klein, S. P., et al., Center for the Study of Evaluation-Leaders Manual, University of California, Los Angeles, 1971

Mathews, W. M. and Sullivan, K., "Public Perceptions of the Computer Impact on Society." AEDS, Vol. 12, #2, pp. 76-83, Winter, 1979.

Orlicky, J., The Successful Computer System, McGraw-Hill Book Company, New York, pp. 98-99, 1969.

109

DESIGN ALTERNATIVES FOR CONVERSATIONAL CONSULTATION SYSTEMS

Thomas P. Kehler, Ph.D.
Mike Barnes
Texas Woman's University

1. On-Line Help Systems in an Educational Environment

The academic computing environment which utilizes a general services timesharing computing resource has a distinct problem in communication with users. A variety of computing resources exists on a system with users of diverse backgrounds accessing the system from various locations. Thus it is frequently the case that a new user obtains an account, logs on, and then hopes to go on to use a document processing, data analysis, or modeling facility on the system. Though published documentation and user-oriented workshops may precede such an experiment, the "no substitute for hands-on activity" usually results in the user accessing the system in need of some on-line help and consultation on how to use the particular piece of software or application programs.

In the particular case of Texas Woman's University, several factors have influenced the development of on-line consultation:

- a new user community - T.W.U. went from effectively no local computing capabilities to an on-line timesharing system with 50 access ports at four different campuses, with general programming, statistical analysis and modeling capabilities in the last two years.
- geographical spread - there are three major locations (Denton, Dallas and Houston) with one city having two locations separated by several miles.
- requirement for low staff costs - the academic services area of the computing center is maintained with a minimum of people (2 full-time) due to budgetary constraints.
- diversity of services - new software, changes to existing software are frequent and require frequent documentation changes.

It is likely that many other computing facilities in a general timesharing environment would experience similar constraints on maintaining a well informed, satisfied user community at a reasonable cost.

The main objective of a consultation system is to provide a dynamic, system-based interface between prepared on-line documents (such as a BASIC or statistical package manual) maintained by a system documentation specialist (SDS) and the end user. Particular objectives for a consultation system are:

- it must be flexible and have an easy to use interface for the user.
- it must provide a facility for keeping documentation up to date.
- it must use unprepared source documents, e.g. the same manuals or documents normally printed for the user.

In the remaining sections of this paper we will consider existing Help systems in light of the objectives proposed. In addition, a basic design for a proposed HELPME system will be given.

2. Overview of Methodologies for Consultation Systems

2.1 Help Files - Non-Indexed Files

The non-indexed file system consists of a number of small files with descriptive names (usually the names of commands or programs). The user can then, either directly or under control of another program, type the entire contents of any particular help file at his terminal. These files usually contain a brief summary of commands and their associated parameters. Very rarely do these files go into depth on a particular subject, due to the fact that long files in such a system are extremely undesirable.

A particular example of this type of system is the system level help file system provided on the DECsystem-20. An area on disk is provided for the SDS to place a variety of files on programming languages, utilities or application systems. To obtain help on BASIC, for example, one types HELP BASIC causing the entire file to be typed out on the terminal.

Three main advantages are inherent in such a system:

- while primitive, it is better than no system at all.
- it is trivial to implement, requiring little more than a program which can type a file.
- it is relatively easy to maintain.

The disadvantage of such a system is that it offers no user interaction and no ability to tailor its responses to individual requests.

2.2 Tree Structured or Menu Oriented Help Systems

Help systems have been designed which provide a menu approach to consultation. Files are organized as nodes of a tree. The user starts with a 'general' node and works to a 'specific' node. One such system is the INFO system in the EMACS editor developed at MIT.[1] A partial implementation of INFO for a general user consultation system was done. The "top node file" of the system is shown below:

```
*node-marker*
File: Dir       Node: Top   UCC.TELL
                                 Info Tree

This is the directory node for  UCC.TELL
Information system.  This system will be
a test system and will change from time to
time.  To exit type  C C.  For help type
H or a ?.

* Menu:

* Guide:      (Guide) Command List
```

```
*Programming Systems:   (Progsy)Prog
*T.W.U. Facilities:     (Facilities)Guide
                                    List
*Terminals:             (Terms)Top
*Editors:               (Editors)Editor
*Document Processing:   (Doc)Top
*Applications Systems:  (Applic)Applic
*node-marker*
```

The file begins and ends with a special character, indicated here as a node marker. The second line of the file gives the file name, the name of the top node and any description. The items in the menu are displayed with the name in parentheses being the file name of the related documentation. This is followed by the top node name of the related file.

The user selects one of these items and is then given a specific block of text of another menu. If, for example, the item programming systems was selected, a new menu of particular languages would be displayed (as stored in the file name Progsy). Further movement down through the tree leads to specific information on one of the programming language facilities available.

This type of system has the advantage of producing a general overview of information available, something often missed in help messages. However, the system can also become very tedious to use if one spends too much time "climbing through the tree", hunting for a particular item. The user is constrained to see the document in terms of the "tree builder's" conceptions.

One feature of extreme importance associated with tree structured documentation is ease of maintenance. An editor which can perform simple tree maintenance, such as add a page or delete a page, greatly increases the system's usability. A disadvantage of the system is that it requires a specially prepared on-line version of the document.

2.3 Category Indexed File

Another method commonly used is that of a category indexed file. This consists of a single file with pointers to various portions of the file. The pointers are either contained in a separate file or built into the accessing program. One advantage to this system is that the source document need not necessarily be specially prepared for use with the on-line consultant, but may be adapted with relative ease from already prepared documentation. If the source document is divided into many distinct divisions, such as a list of commands, then preparation consists of preparing an index containing a string for each section as well as the starting and ending positions of each section. A particular example using the GUIDE program is shown below:

```
GUIDE INFO (ON TOPS20 COMMAND) ? one of the
following:
 ALGOL        APPEND        ASSIGN...
        (lists all commands)

GUIDE INFO (ON TOPS20 COMMAND) LOAD (ON SUB
                                JECT)FUNCTION

The LOAD command places one or more object
files in memory, forming a runnable pro-
gram.

GUIDE INFO (ON TOPS20 COMMAND) LOAD (ON
                               SUBJECT) HINTS
Hints

  Having loaded a runnable program, you may
  then start it (by giving the START com-
  mand) or save it on disk (by giving the
  SAVE command).

  The INFORMATION (ABOUT) MEMORY-STATUS is
  useful for checking the size and place-
  ment of your program in memory.
```

A disadvantage of GUIDE is that the index is coded into the program, making document maintenance impossible. A general disadvantage of the category-indexed method is the degree of familiarity required of the user in interacting with GUIDE. GUIDE functions well in the mode of reminding experienced users of particular items within certain commands. Topical interrogation or cross referencing of subjects is beyond the scope of this method.

2.4 Linked-Index File

The linked-index file system offers great versatility in responding to specific user queries while using an already prepared source document for its information base. The system uses this source document as well as an index to aid the user in finding desired information. Two programs are involved. The first program produces an index of the document. This index is produced through an interactive program which creates a concordance of significant words in the document and then produces an index containing a list of all significant words, as well as a list of references for each word. This interactive indexing program is run by a person familiar with the document who can aid in selecting the significant words. The index produced also contains auxiliary links for each reference which assist in determining the context in which significant words may be found.

The second program is run by the user seeking information and allows him to query the document. Essentially, the user chooses significant words, or groups of significant words, and is provided with small portions of the source document such that he or she can determine the correct location of the information he or she is seeking. Once the user has found the location of what he or she desires, any portion of the file near that location may by typed out.

In a linked-indexed interactive consultant system, the user may specify keywords in a number of ways. The user may input a keyword and ask which available keywords are alphabetically similar to the one specified. The user may also have keyword sets -- groups of keywords which must occur within a certain "distance" of each other in the text in order for a match to occur. For a particular instance of a keyword, the user might wish to know which other keywords appear close to the given keyword (this might help the user in formulating keyword sets mentioned above). The user also might wish to use a keyword in context (kwic) approach, where for each incidence of a given keyword, the sentence in which it appears is shown. The variety of methods by which the user may query the document aids in helping the system to achieve its major goal -- to provide the desired information in the quickest time possible.

The linked-index method is one proposed by the authors for the design of the HELPME consultation system. This design is discussed in more detail in a later section.

2.5 Eliza-Type

Shapiro and Kwasny at Indiana University have implemented an interactive help system based on Weizenbaum's Eliza program.[2] This system accepts natural language sentences from a user concerning various aspects of the operating system. Keywords are stripped from the input sentence and pattern matched against internal patterns. If a pattern match occurs, a specified portion of the script is output.

The primary disadvantage to such a system is that for any but the most trivial of applications, production of adequate patterns and scripts becomes a major project. An extremely large number of user inquiries must be anticipated and written in the script. Also, there must be some means of script maintenance as changes occur.

2.6 Knowledge Driven Systems

One of the concepts introduced in the Eliza-Type system was that of "scripts" for understanding from user input in natural English. Though the Eliza-Type system does not provide a general approach to consultation systems, it does include the very important aspect of a structural scheme for controlling interactions with the user, enhancing the ease of use of the system.

For a system to effectively provide domain-specific information as represented in the documentation and in the user expectations, a "world model" scheme must be employed. A variety of "expert" systems have been developed which consult with a user on a particular subject matter. The SOPHIE system for example consults with a user on details of diagnosing problems in the functioning of an electronic instrument. Domain-specific approaches, though they require significant investigation in the development of representation schemes, are the strongest documentation "interpreters" for users.

3. Linked-Index Structures

An experimental on-line consultant using a linked-index approach has been proposed by the authors. This system, called HELPME, is designed to help answer the following questions:

- What are the advantages of a linked-index approach to an interactive consultant?

- Which index data structure is most effective in terms of flexibility, ease of implementation, and retrieval speed?

- What types of information do users wish to obtain from on-line documents? How does this vary among different classes of users (for example, users unfamiliar with a document versus experts)?

- How does one approach the problem of updating an index in response to changes made to a document?

While HELPME is an experimental system, it must operate in a production environment. Because of this, attention must be directed into making such a system easy to use, both for the index preparer and end user. Also, index format must be such that entries may be quickly transversed while keeping the number of associated pointers to a minimum.

The data structure of the proposed system consists of two entry types. The first type, WORD.BLK, contains the word being indexed (the term 'word' is used loosely and may consist of multiple words) a pointer to a list of occurrence blocks for this word (OCCUR.BLK.PTR), and a number representing the total occurences of this word in the index. The second type, OCCUR.BLK, contains information on each specific instance of an indexed word in the document. This information includes the location of the word in the document, a pointer to the next OCCUR.BLK (if any), pointers to the OCCUR.BLKs of the words in the index which immediately precede and follow the current word in the document, and a back pointer to the WORD.BLK associated with the current word. This structure can be represented graphically as follows:

```
word.blk
-------------------------------------------------
                 index.entry.name
-------------------------------------------------
     number.occurrences       occur.blk.ptr
-------------------------------------------------
```

```
occur.blk
-------------------------------------------------
          byte.ptr        next.occurrence.ptr
-------------------------------------------------
     previous.word.ptr      next.word.ptr
-------------------------------------------------
       word.blk.ptr
-------------------------------------------------
```

- index.entry.name -- string containing word being indexed.

- number.occurrences -- number of occur.blks associated with this block (i.e. number of times this word occurs in document).

- occur.blk.ptr -- pointer to the first occur.blk for this word.

- byte.ptr -- pointer to byte in the document which contains the first character of the current word.

- next.occurrence.ptr-- pointer to next occur.blk for the current word.

- previous.word.ptr -- pointer to the occur.blk of the last indexed word in the document before the current word.

- next.word.ptr -- pointer to the occur.blk of the next indexed word in the document after the current word.

- word.blk.ptr -- back pointer to the word.blk associated with this block.

In operation, the data structure works as follows. All of the WORD.BLKs are stored sequentially in alphabetical order. Not only does this decrease access time, but also allows the user to scan the WORD.BLKs to aid in choosing keywords. The order of OCCUR.BLK storage is not critical, since all association is done through links. The index then consists of a table of WORD.BLKs, each one the header of a list of OCCUR.BLKs. Therefore, by scanning a list of OCCUR.BLKs, each occurrence of a particular indexed word may be located. The user may employ this information in two ways. First, he or she may wish the text surrounding each occurrence of a word to be output in a key word in context (KWIC) format. Second, this set of occurrences can be stored and compared with another set of occurrences of a second word to determine if there are any occurrences of both words appearing close together ('close' is defined in terms of a character distance and may be altered by the user). Note that any number of sets may be stored (within implementation restrictions) and compared with other sets. Another feature of the linked-index system is that for any occurrence of a word in the index, those words which are indexed and surround (within the document) another indexed word, may be quickly determined (by use of the PREVIOUS.WORD and NEXT.WORD pointers).

The format of the data structure can be seen to have one goal -- to allow the user to find a desired portion of the document. This is achieved by:

- specifying groups of keywords which must be located within a certain distance of each other in the text.

- printing portions of a document surrounding the occurrence of a word.

- printing those words in the index which are located near a specified word.

- allowing the user to print any portion of the document he or she wishes, both to aid in searching and to allow the user to read the desired section once it has been found.

Disadvantages of the linked-index consultant include the complexity of both the indexing and user interface programs. Also, for large documents, index size can be several times larger than the original source document. The size of the index also requires that special methods be used in accessing it. One technique used is to keep the table of significant words in core, along with a small portion of the list of references, while keeping the majority of the index on disk. One other major disadvantage of the linked index system is what happens when the source document is updated. Making even a trivial one character change near the beginning of the document would throw off the index pointers and make the index unusable. Because of such problems, there might be a tendency to not update indexed documents because of the labor involved in recreating the index. This is a serious drawback and work should be done investigating various updating schemes.

Because of the flexibility inherent in the linked-index consultant, extensions to such a system would be relatively straightforward to implement. Examples of such extensions include:

- multiple indexes

- addition of tree or category-index access methods to allow information retrieval of a more general nature.

- natural language front end.

Multiple indexes would allow several indexes to be associated with a single document. This might prove useful when a document is used by differing classes of users. The linked-index consultant could then automatically choose the index most suited to the needs of the user. These additional indexes need not necessarily be of a linked-index format. They could contain pointers from other accessing methods (such as category-index and tree-index) so that, in effect, these other systems are integrated with the linked-index system. Since all indexes are stored separately from the source document, any number of indexes can be used to describe a document -- giving the consultant system greater flexibility. In addition, since a linked-index consultant basically processes words given it by a user, it would be possible to add a natural language processing front-end. This front end would accept and parse natural language requests, resulting in words which would then be processed by the linked-index consultant.

4. Summary and Conclusion

Each of the systems reviewed provides potential approaches to developing a consultation system that meets the requirements of an interface between the user and the computing resource which is easy to use and maintain. The Help file system is the most commonly used but lacks in providing a facility which can answer topical questions on resource capability or other subject based items. The tree structured and category-indexed approaches as demonstrated in the INFO and GUIDE systems provide a basis for developing a more sophisticated consultation system beginning with the linked-index HELPME design. In HELPME, the flexibility of indexed references are explored to provide cross-referencing by subject and context specific treatment of terms. The following components for an effective consultation system are proposed:

- A structure containing the logical organization of the subject presented in the document (including the concepts and basic semantics of the area). A minimal approach to this is provided in the INFO tree system. Exploratory relationships among content words in a document can be obtained in the linked-index system as described.

- A set of scripts for responding to various query types. Here the Eliza-Type system does establish the value of scripts for various types of user inquiries.

- A monitor which records the acceptability and performance of the consultation system. No help system currently exists which provides this function of evaluation of the consultation utility. Since an effective consultation system must be tunable to user community needs, such a feedback mechanism for the systems documentor is needed. Historical profiles of user interaction could be used in developing a more effective consultation system.

Implementation of the linked-index system is underway. The initial area of application of this system will be the documentation on the various statistical packages currently available for academic users on the DECsystem-20 computer at Texas Woman's University.

-REFERENCES-

"Interactive Consulting Via Natural Language", by Shapiro, Stuart C. and Stanley C. Kwasny. CACM, August, 1975, 18,8.....p.459

[1] Ciccarelli, E., "An Introduction to the EMACS Editor", MITAI Memo, No. 447, January, 1978.

[2] Shapiro, S.C. and Kwasny, S.C. "Interactive Consulting Via Natural Language", C.A.C.M., 18,8 (August, 1975), p. 459.

NEW ADMINISTRATIVE SOFTWARE FROM DATA BASE TECHNOLOGY

Daniel K. Kehoe

Quodata Corporation
Hartford, Connecticut

The purchase of a mini-computer is by far one of the most common cost-cutting moves made by an educational institution today. A tremendous amount of computer power packed into an affordable system has been the impetus behind locating mini-computers in school environments which traditionally cannot afford large expenditures on hardware and data processing salaries. Most schools delving into computers during the last decade have looked to service bureaus for administrative computing when tight budgets and personnel requirements made in-house computers prohibitively expensive. Most importantly, service bureaus have had the best administrative software available, ready to run and at the right price. As anyone who has made the jump to an in-house computer knows, after the initial hardware investment, an intolerable amount of money seems to be spent on software development and program maintenance.

Certainly, the advent of the mini-computer has done much to make the initial investment in hardware a less painful one. Unfortunately, the cost of producing software for an administrative application is still excessive. Now that mini-computers have found their way inside the school walls, they are expected to perform all the tasks that the larger mainframe system was doing at the service bureau. What this means is that programmers in school environments are being asked to produce relatively much more, given a smaller machine. The cost of producing a line of program code is really the same, whether it be on a mini- or mainframe computer, in terms of programmer effort. Thus, the cost of developing administrative software remains a financial nightmare for most educational institutions.

In mainframe environments, the solution to this dilemma came about with the birth of Data Base Management Systems (DBMS). The goal of DBMS was to cut the cost of developing new applications by using a common, well designed data base structure; a new concept of sharing data and programs to avoid duplication and utilizing easily changeable program modules. It is safe to assume that what worked for mainframe computers will also work for mini-computers, but probably on a smaller scale. A few scaled-down versions of DBMS have been popping up on mini-computers. This prompts us to imitate the progress made in the mainframe environments.

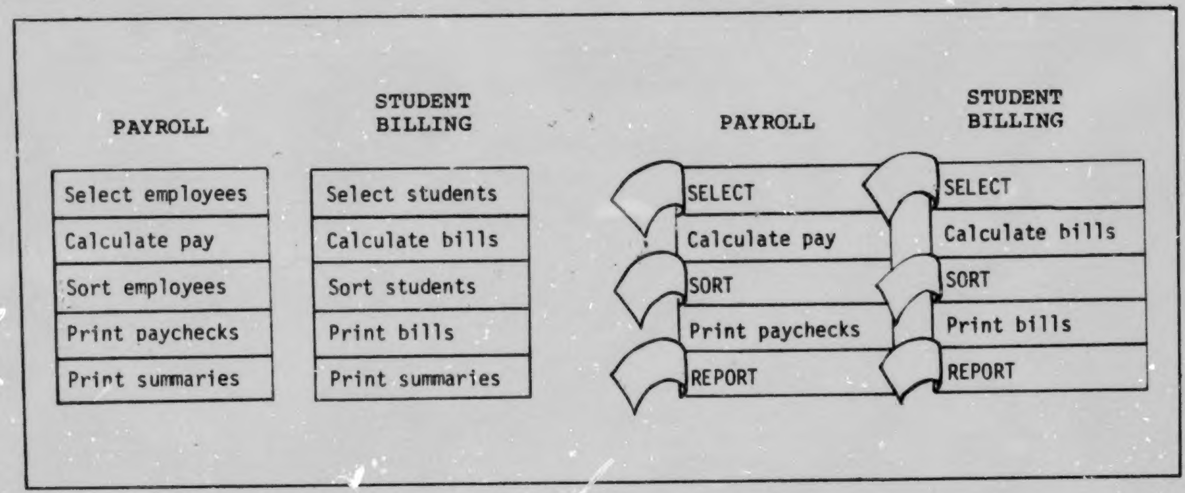
Since programmers can not physically be pushed much over the 100% effort mark, it would be advantageous to magnify the usage for each line of code they produce and to diminish the duplication of programming effort wherever possible. These are our ultimate goals, which represent the best utilization of both the machine and the programmer.

To avoid a lengthy discussion of just what is DBMS, suffice it to say that DBMS offers the same service to data on the computer as a filing cabinet does to information on paper. Simply, it stores it efficiently, retrieves it swiftly and doesn't lose track of it. Although often looked upon as a glorified information retrieval system, DBMS provides integrated sorting, selecting, and reporting modules which are general enough to be pulled out and used by a programmer. It is through utilization of these modules that we can realize our goal of increased programmer productivity.

Consider two application software packages typically scheduled to be developed and run on a school computer — payroll and student billing. Let us first assume that 10 modules comprising these two simplified applications have to be written from scratch, costing 10 units of programming effort (Figure 1). After peeling off a veneer of code from several of these modules, we realize that certain tasks are, in reality, the sort, select, and report modules available to us from the DBMS!

Figure 1

Two Applications and Their Modules


PAYROLL
STUDENT BILLING
Select employees
Calculate pay
Sort employees
Print paychecks
Print summaries
Select students
Calculate bills
Sort students
Print bills
Print summaries
PAYROLL
STUDENT BILLING
SELECT
Calculate pay
SORT
Print paychecks
REPORT
SELECT
Calculate bills
SORT
Print bills
REPORT

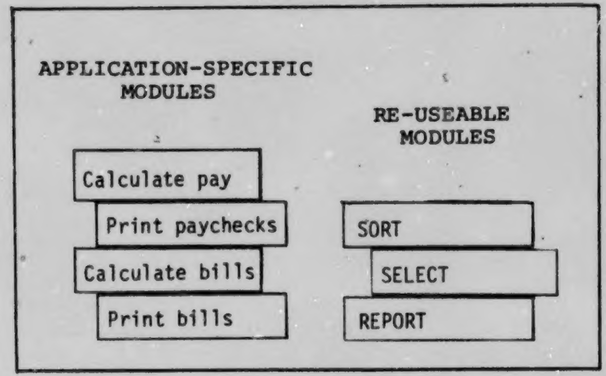
We must realize the differences between traditional program development and this new DBMS-based development scheme. First, this DBMS method relies heavily on modular programming. Every task, whether it be printing payroll summaries or billing students, can be broken down to a series of simplified tasks such as sorting, selecting and reporting (printing). Though this process may appear to be the long way to producing a program, the programmer will eventually see that many of the 10, 20, or even 100 modules defined have already been written. Many of these modules can be very specific, and yet highly re-useable. For example, how many times does the average programmer write an input routine to handle user responses at the terminal? An input module, whose only task is to decipher terminal input, can be used over and over again in virtually any program.

Secondly, a DBMS-based application is for the most part pieced together and not programmed. The modules that can be re-used from other programs are assembled first, then the gaps that are left are filled by programming the needed modules.

We can now talk about two types of modules: application-specific and re-useable (Figure 2). Application-specific modules are coded by the programmer because they perform very specific tasks (such as calculating wages from complicated tax laws). These modules include tailored prompt and help text specific to the application in question. On the other hand, copies of re-useable modules are simply "plugged-in" where needed in the application to perform generalized tasks (such as sorting). These re-useable modules are fine-tuned yet versatile. The programmer's job has now been reduced to supplying these modules with the necessary input and applying the resulting output towards the workings of the next sequential module. Programming effort drops from the original 10 units to 4 units. This includes a small amount of effort needed to "plug-in" these re-useable modules with the application specific modules.

Figure 2

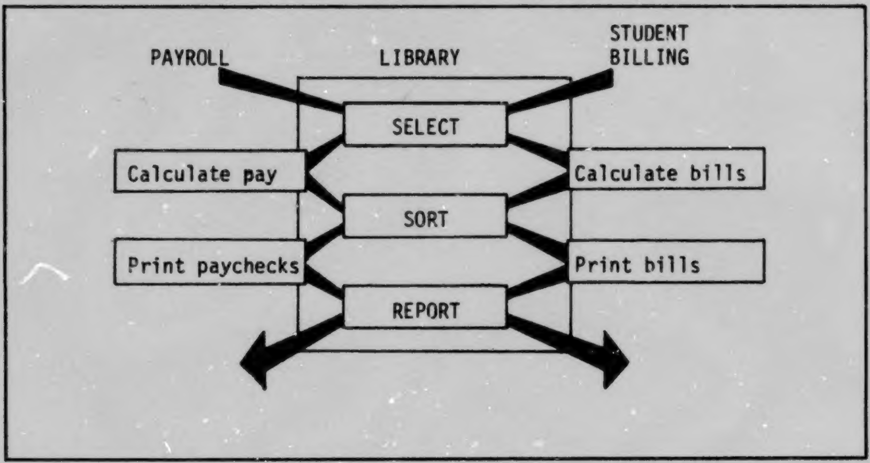
The Two Types of Modules


APPLICATION-SPECIFIC MODULES
RE-USEABLE MODULES
Calculate pay
Print paychecks
Calculate bills
Print bills
SORT
SELECT
REPORT

In more sophisticated programming environments, these re-useable modules can be stored in a "library" on the computer, which is accessed by our two applications as needed. Thus, we can obtain 10 units of application power by storing only 7, thereby reducing the demand on computer storage resources (Figure 3).

Figure 3

Applications Using Library Modules



The advantage of a DBMS-based application after it is finished and running must not be overlooked. Improvements and corrections to the applications are performed by fine-tuning the modules. This is done by making enhancements in speed and technique that will not alter the flow of the application. Additionally, as problems arise and are corrected in one re-useable module, it is a simple matter to replace the old module with the new module in both our applications. This, in effect, multiplies program support capabilities without increasing support effort.

Finally, DBMS's disciplined data file structure minimizes interfacing problems and its standardized command language encourages the use of one set of commands for all applications. The commands to retrieve a billing record are of exactly the same syntax as those to retrieve a payroll record. While each application is self-contained, their DBMS-based construction supports smooth inter-communication, allowing relevant data to be shared back and forth. Since all the applications have the same data file structure, DBMS can be called in to sort, select, or report on the data where the applications' abilities leave off. For example, the ad-hoc reporting of DBMS can be used to produce an answer to a "what if and how much" question posed by a user about granting a pay raise to all employees. While this is a typical question, it is unlikely that a payroll application package would include a mechanism necessary for handling this question. DBMS can easily answer such a request using its sort, select, and report capabilities; and taking advantage of the common data file structures between them.

In summation, the benefits of a DBMS-based application software system are:

- The ability to create solid administrative software in less time and at a lower cost having the sorting, selecting, and reporting capabilities typical of DBMS.
- Increased programmer productivity by aiding application development and program support effort.
- Less demand for system resources by decreasing the amount of program code residing on the computer.
- Continuity of operation which minimizes user error and shortens training time.

The presence of DBMS on a mini-computer is unquestionably the catalyst necessary to create solid administrative software, given the time and cost restraints prevalent in educational institutions. By multiplying the effective output of programmers, DBMS on a mini-computer becomes an effective cost-cutting tool.

DESIGNING
EDUCATIONAL MICROCOMPUTER SOFTWARE: A MODEL IN USE

Marge Kosel
Instructional Coordinator
Minnesota Educational
Computing Consortium

Software development for microcomputers is a growing commercial enterprise, a consideration for school administrators, and a concern for classroom teachers. The Minnesota Educational Computing Consortium (MECC) began developing microcomputer software in the fall of 1978. Although MECC staff was experienced in developing software for a timeshare system, the microcomputer offered new problems and a new challenge; screen layout, graphic techniques, and the role of sound opening new possibilities. This past year, guidelines and techniques were created to facilitate and enhance the software development process for instructional computing.

The development process divides into six stages:

1. Analysis   Analyzing the curriculum and teacher goals, and determining whether the computer is the means to achieve those goals.
2. Preliminary Planning   Researching the concepts to be taught, defining the objectives for each lesson, and assessing the computer's ability to present the material.
3. Design   Defining the order of material to be presented and the interaction of computer and student.
4. Screen Layout   Creating a frame-by-frame layout of material to be presented.
5. Programming   Writing the code for the computer.
6. Testing   Evaluating the software results by professional staff and by use with students, and re-working the software based on those evaluations.

Note: A glossary is provided at the end of this article for a definition of terms as used herein.

This paper details the first four steps in the development of microcomputer software: analysis, preliminary planning, design, and screen layout.

Analysis

The ideal way to develop microcomputer software would be for instructors to define their goals, to look at the teaching techniques for obtaining the goals, and to judge which computer features will add to the presentation of the material. However, most software design evolves not according to the ideal, but comes about because the microcomputer is available and it is felt it should be incorporated into the curriculum. Such incorporation should be carefully examined. It is a mistake when the microcomputer is allowed to perform tasks that can be done better, or as well and perhaps less expensively, by other means. If students can perform a science experiment and see the results, the computer is not an effective replacement for the experiment. If a ten-question social studies worksheet presents questions in the same order as the computer, even though the computer provides an immediate response it is not a cost effective way to drill students.

The microcomputer should be viewed as a teaching tool and used in the curriculum when it can create new learning experiences or present a fresh and useful approach to learning.

There are characteristics of the microcomputer that do add to the presentation of material. Computers can randomize problems and make ordinary drills interesting, allowing students to repeat drills without receiving the same set of questions. In the mathematics classroom an important skill is to answer or estimate a problem quickly. A teacher cannot easily time students on their individual responses. The computer can time both accurately and efficiently. Computers also calculate quickly, allowing students to look at higher level concepts without getting lost in the internal mathematics.

The ability of the computer to calculate quickly and to do mathematical modeling allows the development of simulations in other subject areas. The computer can be viewed as a time machine for students to study the stars during the day - or six months from now, travel to Oregon back in the 1850's, or munipulate data collections to analyze voter choices in presidential elections.

Special needs of the student who interacts with the computer may also serve as a justification for the computer in instruction. A handicapped child using a light pen can "control the pencil". Voice input and output individualize foreign language instruction. Color, graphics, and animation demonstrate the moving parts of an engine which otherwise lay flat in the book illustration. With the computer students slow down the action of a machine to observe its different cycles.

Educating is the major goal and only when the computer enhances learning should it be used in the curriculum.

Preliminary Planning

If the decision is made that the computer will enhance the curriculum and goals have been set, research begins. Any major development demands exhaustive research. The scope of the research will in part determine the "portability" of the product. If a product is to be used in one school by one teacher, software can be written to reflect the objectives of a single teacher. However, in a major development where a product is used by many teachers, the product must be made "to fit" many curricula. To make the product "fit" the developer/author must select out of the broader area of investigation those parts to be included in the development. The parts are organized into lessons with each lesson developing one concept. After the lessons are organized a determination is made on how much material will be presented by the computer, how much through documentation, and how much through other media.

In pre-planning the developer should be aware that unless the microcomputer is connected to a printer, the teacher has no evidence that a student has run a specified program. Working into the program design some written interaction such as student worksheets, not only provides variety of experience, but serves as a record for the teacher. Authors not only decide how much instruction is presented by computer or in documentation, but which instructions should be included. Depending on the age of the student, instructions may be excluded from the computer program and teachers directed to give the instructions to students verbally.

In considering the role the computer is to play in the instruction, how graphics are to be used needs exact determination. Most microcomputers provide the ability to use graphic representation or animation, and many authors are concerned about their overuse. This writer is of the opinion that materials are developed to fit the technology of today and graphics are an important part of that technology. We live in a world where students are both taught and entertained by "Sesame Street" and "The Muppets". That the visual affects learning is evident, if not wholly understood. Use of computer graphics in instruction, however, should not be included to entertain but to enhance learning. The creative author will explore ways to include graphics which contribute specific learning.

In building the senses: visual, audio, tactile, and sound into lessons, the developer should evaluate the role of sound most carefully. If a headphone for the individual student can be supplied at the computer to prevent sound effects from disturbing the rest of the class, sound can be effectively built into the program.

Color graphics are not only enjoyable but can be used for reinforcement and to illustrate. Color can be used as a reinforcer for small children in place of or in addition to a word. Blood cells can be illustrated in red and white and bring reality to the program. Color is also used to attract student attention. However, overuse of color will disturb the viewer. Color should not be used randomly without specific educational objectives. In using color with numbers, colors should represent an educational concept, i.e. a place value, digit, etc.

The design of the character set to be used for screen output is another consideration. Should it have both upper and lower case, be in English or German or Spanish? If the material is for elementary students, a larger character set with both upper and lower case may be preferable.

The Design

With earlier steps defined and refined, design of the materials is the next step in software development. Readability is a factor. If a new concept is being presented, experience has shown that the reading level of the text should be one grade level lower than the grade level of the concept to be taught. In this way reading should not interfere with the concept to be mastered. The readability of materials should be tested on a student sample before it is finally adopted.

The developer must also order the material to be presented within a lesson. Material should be organized to allow students to interact with the computer as frequently as possible. Large blocks of text that do not give students a chance to respond to a question should be avoided.

What the student is to type into the computer, "input", is another consideration for the author. For young children, minimizing the amount of typing may be an objective, as is the use of indirect questions. Questions should be explicit. Since microcomputers print very fast, it is better to ask, "Do you want instruction?" than to use a computer term such as, "Info?". Instead of asking students to input more than one item on a line, separate questions. For example, instead of "Type the day, month, and year", ask:

> What day is it?
>
> What month is it?
>
> What year is it?

Error checking, or responding to incorrect input, is important to the design for all computer software. Anticipating what the user might type in and giving clues to the expected response is helpful. Directions should be clear. A question such as "What month is it?" does not tell the student if they should type January, JAN., or "1". A better choice is:

> What month is it? (Enter a number from 1 to 12).

Even now the author must anticipate the kind of error a student may make. A student may type an "L" instead of a "1". A more detailed programming design could automatically accept the "L" as a "1". Another method 'freezes' the keyboard to accept only the "number" keys. Any other key typed will not print on the screen. A key can be programmed to designate a HELP key, or students can be instructed to type HELP if more explanation is needed.

What should the response of the system be for the student who fails to input a response, but instead presses the ENTER or RETURN key? If a math problem is presented, will the author allow students to press RETURN and continue, or will a routine be built into the program to try and elicit a response.

With drills, authors must vary the responses to avoid monotony. Ideally, there will be random selection from a group of responses, without using the same response twice.

Increasing the reading level of positive responses has been suggested as a way to motivate students to read, since students tend to decode new words which praise them.

Screen Layout

The foregoing discusses some of the characteristic problems and challenges to be kept in mind when designing instructional software. The developer progresses through the general design of the material, determines input and output, defines and orders the concepts to be taught, and is ready to decide how the screen layout should appear. A design sheet of the screen should be created for the particular microcomputer the developer is using. Then each screen is designed for the way it should look in the program. The design sheet should be laid out to the character width and length of the screen. Text to be presented is written on the 'sheet' in the exact position it is to appear. The writing can be in pencil with special notes to the programmer in red ink, such as "slow down the speed of the printing" or "only allow a number to be input". Comments are placed at the bottom of the screen and the next screen or screens that will link to it should be noted.

Two general rules to follow in designing a screen are: Anything that is monotonous is bad, and since blank space is free, use it. The developer looking at the screen should take the role of a director of a play or cartoonist animating a movie. Eye contact, movement, and an interesting layout are all important. Pages of text should be varied in format and style. Pages should not all look alike. Scrolling of the screen should be avoided by making sure that all material in a section fits on one screen.

Because it is hard to read, inverse and flashing should be used sparingly. An effective way to highlight may be to control the speed of text to appear on the screen. The eye views the screen from left to right and from top to bottom. This should be kept in mind when presenting material, as well as the fact that the center of the screen is a focal point for viewing. When action is appearing on the screen, it should not distract the focus from the area where the concept to be taught is being presented. Enough time should be allowed for the viewer's eyes to refocus from one set of instructions to another. Sometimes, building in a time delay between one action and the next will help with focusing. Authors should not determine a student's reading speed and present material on a screen which 'clears' before the student can read it. Reading speed should be controlled by the student through use of an instruction at the bottom of the screen such as, "PRESS KEY TO GO ON", useful when the author wants screens changed and does not require input from the student.

The screen layout of a lesson can make the difference between a successful lesson and one that students will resist. Take time to consider the total appearance of the screen. The appearance of the screen is a challenge for authors and not a task to be left to the programmers.

Programming And Testing

The last two steps in the software development process for instructional computing are programming and testing. After the layout of the material has been decided, the programmer codes the software and the materials are tested. The amount of time necessary to test and revise the product should not be underestimated. If the author has done a good job on the design of the material, the effort needed to implement the design will be somewhat reduced, but will still be a substantial task.

Conclusion

The cost of developing quality eductional software is high. Cost calculations for a recent MECC development project for a diskette of seven lessons, including authors, consultants, programmers, testing and documentation, approximated $10,000. Because of the high cost both in dollars and in human energy, software development must be skillfully organized, logically pursued, and constantly examined. Authors must scrutinize each stage of development and be prepared to design and redesign until materials are instructionally sound.

Glossary

Author: A person who develops curriculum material. In this article, "author" is synonomous with "developer"

Data: Information of any kind.

Diskette: A small storage device used on microcomputers. Information can be magnetically stored on a diskette similar to the way music can be stored on a tape.

Documentation: A written description to accompany a piece of software or hardware. Sometimes documentation refers to internal documentation within the coding or program. In this paper it refers to teacher support materials which will assist in the use of computer programs.

Flashing and Inverse: Some microcomputers allow the programmer to print black letters on a white background (inverse) or flash between white or black and black or white.

Input: Information arriving at a device. What the user types into the computer.

Light Pen: A pen-like instrument that will attach to a microcomputer and allow the holder to touch the screen with the pen instead of using the keyboard.

Output: Information leaving from a device. In this article, output is information which would appear on the television screen.

Program: A sequence of instructions given to the computer. Describing a process.

Programmer: A person who writes a program and codes the software.

Screen: A surface on which information is displayed, such as on a television screen.

Scroll: To move all the text on the screen, usually upwards, to make room for more text.

Software: As used in this article, "software" refers to the material or lesson presented on the computer.

Text: In this article, "text" is the writing that will appear on the screen.

Voice Input: Many microcomputers now have the capability for the attachment of a microphone for "talking to" the computer. The computer can be "taught" to understand a certain number of words.

Voice Output: Some computer systems can now project a voice and "talk". Although this speech is mechanical and limited, a more refined capability will soon be available.

Other References

A Guide To Developing Software For The Apple II, MECC Instructional Staff, January 1980

A Programming Style Manual For SESAME PLACE, Joyce Hakansson, Dennis Sullivan, Children's Television Workshop.

Developing Microcomputer Software, Kent T. Kehrberg, Minnesota Educational Computing Consortium, Paper for NECC Conference, January 1979

SESAME PLACE Style Manual. David Ahl, Jane Berentson, "Oregon Computing Teacher", September 1979

A COLLEGE LEVEL INTRODUCTORY COURSE ON COMPUTERS IN THE CLASSROOM: A CASE STUDY

James J. L'Allier

and

Robert D. Tennyson

Instructional Systems

University of Minnesota

A College Level Introductory Course on Computers in the Classroom: A Case Study

Abstract

The purpose of this paper is twofold: To identify instructional design variables directly related to the development of computer-based instruction (CBI) and to show how these variables can be applied in the actual development of a course. The course, "Computers in the Classroom," is a computer literacy course that is not only concerned with the teaching of a specific content but with the attitudes that students have about that content as well. The course attempts to dispel many of the common misconceptions held about computer technology by exposing the student to positive experiences with the computer. Related to specific overlapping course objectives, these experiences consist of three components: Hands-on work with the computer in a CBI mode; films, speakers, and demonstrations; and supplementary readings. A basic structure of these key components is given so that the reader can design or modify a similar course.

A College Level Introductory Course on Computers in the Classroom: A Case Study

Any course is an artifact; an artificial structure that enables the student to deal with an unknown exterior environment. Ideally this artifact is designed in accordance with the internal goals and purposes of the student. In other words, a good course is an interface between what is to be known and what the student perceives as being important enough to know (Simon, 1969). For the most part, these perceptions are shaped by previous experiences with the subject matter of the course, along with the positive or negative feelings associated with those experiences (Bloom, 1976). Such feelings, or subject-related affects, are powerful factors in goal-shaping behavior, positive experiences giving rise to goals of subject mastery for the sake of interest growing out of enthusiasm; negative experiences promoting goals of subject toleration for the sake of getting through an ordeal. What is noteworthy about these experiences is that they can be either real or the outgrowth of a mythology based on misconceptions about the subject--misconceptions that are vicariously interpreted as real experiences. It makes absolutely no difference if the "experience" is part of the student's mythology--founded on hearsay, misconceptions, and ignorance--or is a real experience, based on actual and verifiable fact. What *is* important is that these perceptions, real or otherwise, do form a predisposition that all students bring to any content area, a fact that must be taken into account in the design phase of all courses--our course, "Computers in the Classroom," being no exception (Tennyson, 1975).

From the onset, what we wanted was a course that would dispel many of the common misconceptions that currently surround computers and their capabilities; misconceptions that have tended to form an elaborate mythology based on an image promoted by a 20th centure Ludditeism perpetuated by many who fear this and misinterpret technology (Tennyson & L'Allier, in press). In the course this "de-mything" process comes about through the exposing of the student to positive experiences with the computer, experiences intended to alter those attitudes that could impede learning and promote a negative goal-shaping behavior. It is only after this "affective priming" that the task of teaching the content portion of the course would be broached--a task concerned with the non-technical and low-technical aspects of the capabilities and limitations of the computer, and of its social, vocational, and educational implications (Moursund, 1976).

Affective Concerns

As this is an introductory course a non-computer background is assumed--a point that is stressed in the catalogue description and by the instructor during the first meeting of the class. Our purpose here is quite simple, to allay pre-course anxiety regarding any lack of technical knowledge in this area.

Another strategy that is used to reduce initial student anxiety is the employment of a mastery model for all tests. In this model a student, who fails to attain mastery upon taking the original computer-based test, can review the appropriate materials and re-take another computer-based test until mastery is achieved.

Along with the mastery model the availibility of the instructor and computer lab assistants for technical aid is, in the design of this course, considered necessary to relieve the frustrations and answer the questions that all new users of this technology have. For example, nothing is more frustrating for new users than to get system error messages and not know what they are doing wrong.

Underlying all of these positive affect-building measures is the assumption that computer technology is an anxiety provoking field and that anyone new to it will, to varying extents, have personal misgivings and feelings of technical inadequacy regarding it. All aspects of "Computers in the Classroom" rest on this assumption and, to the end of creating a positive attitude toward this technology, the readings, demonstrations, lectures, and mastery tests are selected and implemented on their ability to dispel the negative predispositions typified by the new user.

Course Context

Before an attempt is made to describe the course, it should be viewed in its academic context as this is the initial environment that shaped the course. First, "Computers in the Classroom" is one of a series of five one credit courses in instructional technology that forms an introductory core for students enrolled in either a masters or doctoral program in Instructional Systems at the University of Minnesota (Tennyson, 1976). Second, this course is also a certification requirement for the Media Generalist Certificate in Minnesota; students in this category being either librarians or audio-visual personnel seeking to up-grade their positions. Last, many advisors recommend that "Computers in the Classroom" be included in their advisees' plans of studies when their programs contain a computer emphasis or component. Students representative of this latter category have come from the fields of curriculum, computer science, and health sciences. What emerges from these categories is a very diverse student profile to which both the affective and subject matter components of the course are tailored.

The Course

"Computers in the Classroom" is a one credit course that is an introduction to the use of computer technology in instruction. It consists of classroom, self-intructional, and computer managed modules. Because it is a one credit course, there are only 10 class meetings, with the student being expected to put in an additional 20 hours of outside study. Much of this additional time is spent on the completion of the four modules that make up the course content. Each of these modules have specific objectives which are:

MODULE NO. 1:

The student will demonstrate the ability to successfully log-on, access, and run a series of programs that represent the following instructional types:

A. Simulation
B. Problem Solving
C. Drill and Practice
D. Tutorial
E. Games

MODULE NO. 2:

The student will understand what a computer is, how it works, and its history.

MODULE NO. 3:

The student will become knowledgeable of the impact that computer technology has, and will continue to have, on society--with special emphasis on the use of the microcomputer in instruction.

MODULE NO. 4:

The student will be able to identify the components that make up a time-share computer system, along with demonstrating a familiarity with some common computer terminology.

MODULE NO. 4 OPTION:

For those students interested in pursuing a specific application of the computer in a field of personal interest, Module No. 4 may be substituted. Where this option is exercised, students are expected to demonstrate, either in oral or written form, a knowledge of how the computer is employed in some aspect of their chosen field of interest. All topics are cleared with the instructor by the second week of the quarter.

Overlapping Objectives

Rather than dealing with each objective as a mutually exclusive goal, our preference was to overlap them thereby forcing the student into integrating concepts learned in one module with those learned in another (Merrill & Tennyson, 1977). This intersecting feature is best illustrated by a Venn Diagram (Figure 1).

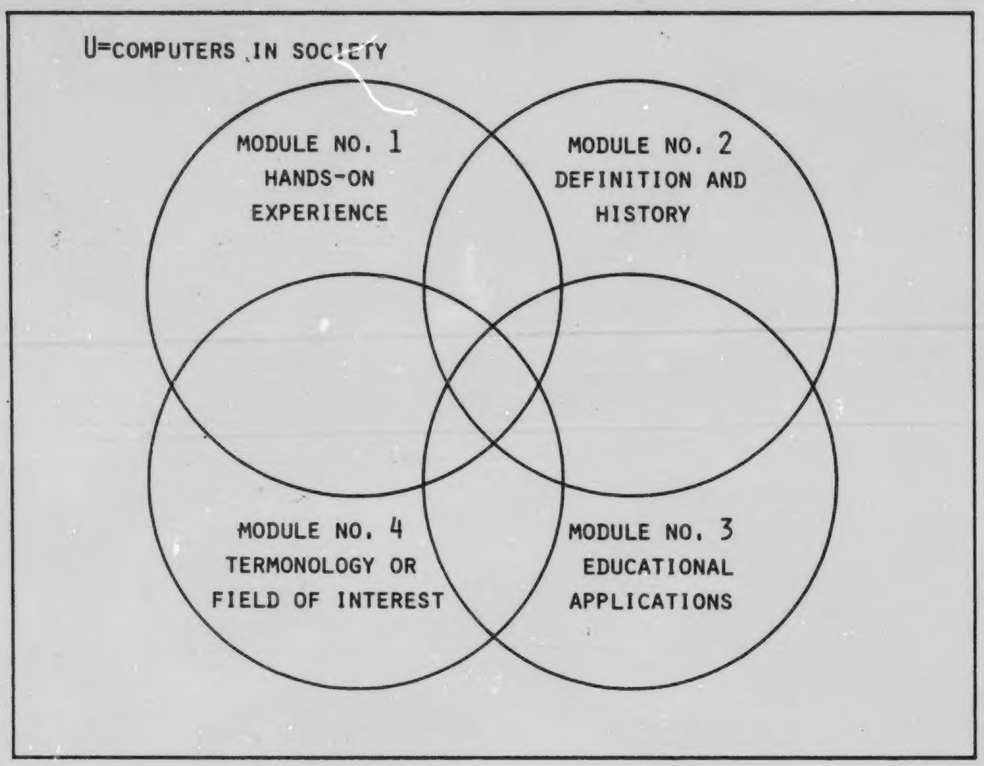


Figure 1. Venn Diagram of course objectives.

If, for example, the student were working on Module No. 3, the overlapping feature of the course's goal stuucture would assure not only a dealing with the educational applications of the computer per se, but with its history and specific terminology as well. Also, the student would be learning about these applications through the actual hands-on running of representative educational program types.

Again, this overlap occurs with any of the goals of the course—our major purpose being a meaningful integration of concepts rather than a fragmentation of isolated experiences.

CBI Component

Not only is "Computers in the Classroom" integrated in terms of its goals, but in its use of the computer as an instructional component of the course as well. Over one-third of the course is administered in a computer-based intructional (CBI) environment. Mastery tests for each module, along with the various instructional programs for Module No. 1, are all accessed via a time-share system.

At the start of the quarter the class is issued a system ID (identification code) and each student personal four digit code, along with procedures for logging on and accessing test materials-prodedures which are gone over in detail during the first two weeks of the course.

When a student feels that the concepts for a particular module has been mastered, he or she logs on and accesses the mastery test for that module by inputting the personal four digit code exclusively used for taking tests. In order to achieve mastery on any test the student has to achieve a score of at least 95 percent (Rothen & Tennyson, 1978). If the score is lower than this, the student is temporarily locked out of the test for 24 hours. After sufficient review, the student can then go back and re-take those items that were not mastered the first time. Figure 2 is a flow-chart of the testing procedures used in the course.

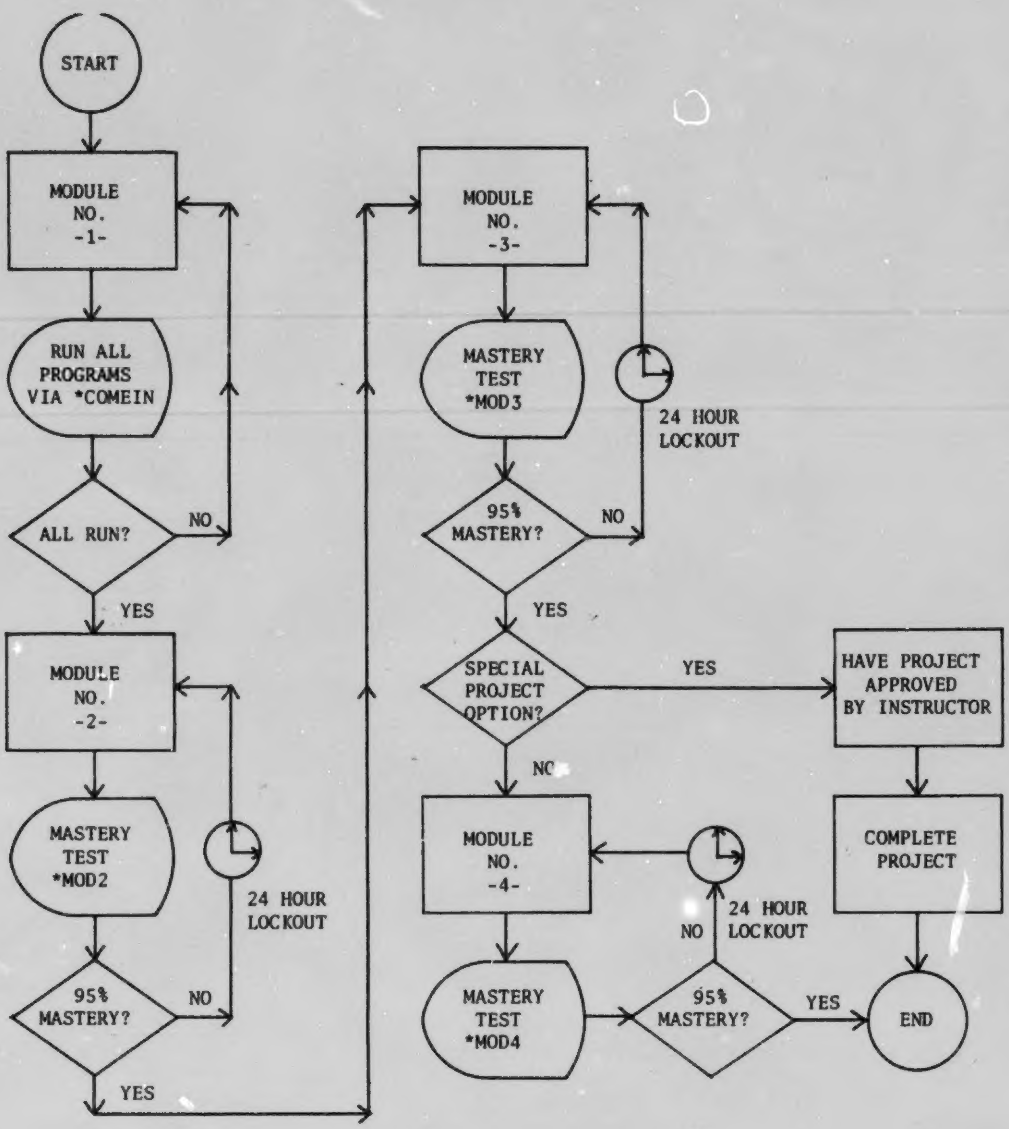
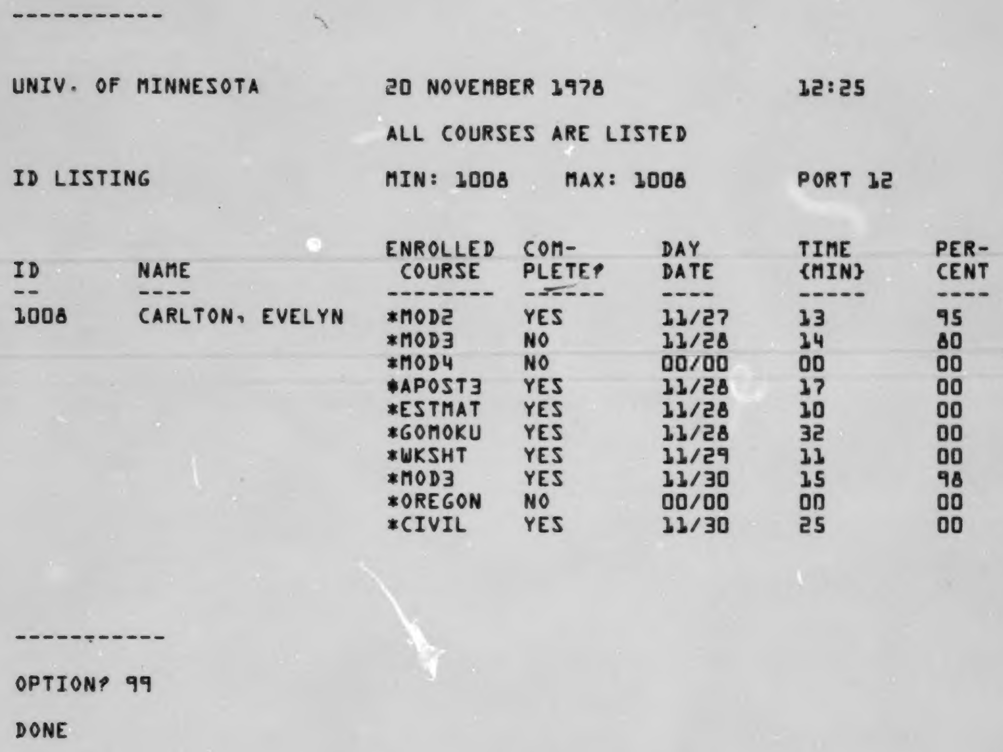

START
MODULE NO. -1-
RUN ALL PROGRAMS VIA *COMEIN
ALL RUN?
NO
YES
MODULE NO. -2-
MASTERY TEST *MOD2
24 HOUR LOCKOUT
95% MASTERY?
NO
YES
MODULE NO. -3-
MASTERY TEST *MOD3
24 HOUR LOCKOUT
95% MASTERY?
NO
YES
SPECIAL PROJECT OPTION?
YES
HAVE PROJECT APPROVED BY INSTRUCTOR
NO
MODULE NO. -4-
24 HOUR LOCKOUT
NO
COMPLETE PROJECT
MASTERY TEST *MOD4
95% MASTERY?
YES
END

Figure 2. Flow-chart of course CMI testing component.

Because tests are accessed on a CRT, and personal codes are used by all students, test security has failed to be a problem. Also, because records are maintaindd each time a student logs on, accesses a test, and runs a program, studying the test rather than the materials has posed little problem. This same record keeping system also facilitates the spotting of areas where students may be experiencing difficulty, along with maintaining accurate student information for grading purposes; grades being assigned according to the number of modules completed at criterion level. A sample of a student report has been included in Figure 3.



```
-----------

UNIV. OF MINNESOTA          20 NOVEMBER 1978          12:25

                            ALL COURSES ARE LISTED

ID LISTING                  MIN: 1008    MAX: 1008    PORT 12

                            ENROLLED  COM-     DAY      TIME    PER-
ID      NAME                 COURSE   PLETE?   DATE     (MIN)   CENT
--      ----                --------- ------   ----     -----   ----
1008    CARLTON, EVELYN     *MOD2     YES      11/27    13      95
                            *MOD3     NO       11/28    14      80
                            *MOD4     NO       00/00    00      00
                            *APOST3   YES      11/28    17      00
                            *ESTMAT   YES      11/28    10      00
                            *GOMOKU   YES      11/28    32      00
                            *WKSHT    YES      11/29    11      00
                            *MOD3     YES      11/30    15      98
                            *OREGON   NO       00/00    00      00
                            *CIVIL    YES      11/30    25      00



-----------

OPTION? 99

DONE
```

Figure 3. Example of a student report from course CMI facility.

Classroom Component

Another very important third of the course is the classroom phase. Made up of a series of lectures, demonstrations, and one film, the content is designed to both supplement and illuminate the course readings and CBI activities. University staff and outside speakers are employed extensively in these presentations, adding their particular area of expertise to the existing course curriculum.

The following is a lecture/demonstration/guest-speaker schedule from the 1979 Fall Quarter.

September 25	Introduction to the course and log-on procedures
October 9	Review of log-on procedures SPEAKER: Jim L'Allier, U of M TOPIC: "A Short History of the Computer: The Economy of Scale" (Two Periods)
October 16	SPEAKER: Sue Talley, T.I.E.S. (Total Information Educational Systems) TOPIC: "GIS and MOIS Information Systems" (Two Periods)
October 23	SPEAKER: Celia Ellingson, U of M Library TOPIC: "Computerized Library Search Systems"
November 20	SPEAKER: Dan Forde, T.I.E.S. TOPIC: "Micro Computers in Education: A Demonstration of the TRS-80, PET, and Apple II" (Two Periods)
November 27	FILM: "The Mind Machines"
December 1	Course evaluation and wrap-up

In so far as this is an introductory course the speakers were selected not only for their unique points-of-view, but for their ability to communicate those views in a clear and non-technical way. In some cases experts were passed over in preference to those who possessed this quality--a judgment that we feel is necessary for this type of course.

Class Communications via the Computer

On an administrative note, speakers cancel and dates and times are changed--this is usual and should be expected. But, when the class only meets ten hours, and not at regular intervals, these routine changes can cause communication problems between the instructor and the students. Our solution was to write a short message program and use the computer to disseminate any schedule change information as the student would have more contact with the computer in the interim than with the instructor. Consequently, each time a student logs-on, he or she is given the schedule for the next class meeting, along with any changes that have occurred. This facility has also proved useful as a paging system for the instructor when there is a need to contact a particular student. The point here is decidely administrative, but to the student it is another example of a computer solution applied to a problem.

Reading Component

Closely integrated with both the CBI and classroom components are the course readings. Constituting the final one-third of the course content these readings were selected on the basis of their ability to deal with a technical topic in a way that was both appropriate and considerate of the students' backgrounds.

Grouped according to the respective objectives for each module, the readings supply depth to the surface concept structures formed by both the CBI and classroom components. The following is a listing of the course readings by module; for full bibliographic information, please refer to the Reference section of this paper.

Module No. 1:
James Worthy. "Perspectives on Education in the 1980's"
Module No. 2:
The Computer Age
John Kemeny. *Man and the Computer.* (Chapters 1,2,3, and 4)
More About Computers
"The Numbers Game"
Module No. 3:
"The Computer Society"
Jessica Davidson. "The Square Root of Tuesday"
Kenneth Edwards. "The Electronic Newspaper"
Alan Kay. "Microelectronics and the Personal Computer"
John Kemeny. *Man and the Computer.* (Chapters 7 and 8)
Selma Marks. "The Ubiquitous Computer"
Robert Smith. "A Funny Thing is Happening to the Library on its Way to the Future"
Hollis Vail. "The Automated Office"
Module No. 4:
"A Glossary of Terms"
Hardware at a Glance
John Kemeny. *Man and the Computer.* (Review Chapters 3 and 4)

Comment on selected readings. A comprehensive work that covers most of the objectives of the course is John Kemeny's book, *Man and the Computer* (1972). In a sense it is our official textbook, with all other readings acting as supplementary up-dates. Published in 1972, *Man and the Computer* is decidedly time-share oriented, a fact that we didn't object to as most educational computer systems in colleges and universities are time-share ones--including the system used for this course.

In areas that are not dealt with in Kemeny's book we rely on supplementary materials from journals and vendor publications. Noteable areas covered by these materials include computer history, micro and mini computers, and word processing. Our criterion for selecting these supplementary materials was that they be readable, authoritative, and current with this fast-paced and ever-changing field.

Evaluation

At the end of each quarter all courses at the University of Minnesota undergo a required evaluation in the form of a general student opinion survey. Subjective as such instruments are, it was deemed an appropriate tool for our course as one of our major concerns was the changing of negative attitudes about computers through positive experiences with them--that is, subject-related affect (Tennyson, 1978). Table 1 is a tabulation of the evaluation from the Fall Quarter of 1978. Only those statements that are content related have been included; items concerning the instructor were eliminated as they are not really germane to the content description of this course.

Table 1

Student Opinion Survey Tabulation

For Computers in the Classroom

Summer 1978 through Fall 1979

SURVEY ITEM	VERY STRONGLY DISAGREE 1	STRONGLY DISAGREE 2	DISAGREE 3	AGREE 4	STRONGLY AGREE 5	VERY STRONGLY AGREE 6	MOST STRONGLY AGREE 7
THE READING MATERIALS--including the textbook:							
ITEM							
1. held my attention easily.	0	0	0	36.7	18.6	17.0	26.7
2. was clearly written.	0	0	1.3	37.7	22.5	24.2	14.1
3. served well its intended purpose.	0	0	0	19.9	24.3	24.6	30.6
THE TESTS:							
ITEM							
4. concentrated on the important points and topics in the subject matter.	0	1.3	1.3	31.9	23.6	22.8	18.2
5. seemed to have been carefully and conscientiously prepared.	0	0	1.1	22.8	21.9	22.6	31.6
6. were about the right length.	0	0	6.2	17.6	24.8	27.0	24.3
7. were clearly worded.	0	0	9.2	21.7	13.7	30.4	24.9
8. seemed to be a good measure of my knowledge and understanding.	1.1	0	6.1	31.7	16.4	18.1	26.6
IN GENERAL							
ITEM							
9. procedures for determining grades were appropriate for this course.	0	0	3.6	11.9	25.4	15.0	44.1
10. the amount of work required was appropriate for a one credit course.	0	1.1	5.5	20.0	19.0	21.9	32.6
11. adequate information about how well I was doing was readily available.	0	0	1.3	11.0	17.0	21.7	49.0
12. adequate help was available when I needed it.	0	0	0	12.0	16.7	18.1	53.2
13. course responsibilities were clearly defined.	0	0	0	3.6	15.0	18.1	63.1
14. the course was an example of how computers are used in a classroom.	0	0	0	14.5	19.5	22.8	42.8
15. I would recommend this course to others in my field.	0	0	6.0	11.0	9.9	30.2	42.7

More about computers. Armonk, New York: IBM, 1977.

Moursund, D. What is computer literacy? *Creative Computing,* 1976, 2(6), 54-55.

Popham, W. J., & Baker, E. L. *Systematic instruction.* Englewood Cliffs, N.J.: Prentice-Hall, 1970.

Rothen, W., & Tennyson, R. D. Application of Bayes's theorem in designing computer based adaptive instructional systems. *Educational Psychologist,* 1978, 12, 317-323.

Simon, H. A. *The sciences of the artificial.* Cambridge, Massachusetts: M.I.T. Press, 1969.

Smith, K. F. A funny thing is happening to the library on its way to the future. *The Futurist,* 1978. 12(2), 85-91.

Tennyson, C. L., & Tennyson, R. D. Evaluation in curriculum development. *Educational Technology,* 1978, 18(9), 52-55.

Tennyson, R. D. Competencies and content necessary for graduate studies in instructional systems. *NSPI Journal,* 1976, 10, 7-10.

Tennyson, R. D.Application of computers in education. *Audio-Visual Instruction,* 1974, 19(5), 49-52.

Tennyson, R. D. Instructional systems: Development, evaluation, and management of learning environments. In *Handbook on teaching educational psychology,* D. J. Treffinger, J. K. Davis, and R. E. Ripple (Eds.). New York: Academic Press, 1977.

Tennyson, R. D. Evaluation technology in instructional development. *Journal of Instructional Development,* 1978, 2(1), 19-26.

Tennyson, R. D., & L'Allier, J. L. Breaking out of the Skinner Box. *ITED Newsletter,* in press.

Vail, H. The automated office. *The Futurist,* 1978, 12(2), 73-78.

Worthy, J. C. Perspectives on Education in the 1980's. *AEDS Journal,* Spring 1977, pp. 60-67.

In terms of the subjective opinions of the students the data in Table 1 is very positive, with the mean for all of the listed items being 4.65; a figure between "Agree" and "Strongly Agree." Also, in more objective terms, all of the students who were enrolled in the course First Summer Session 1978 to the present, have achieved mastery in all four of the course modules--a measure that we feel is the final and most important judgement concerning the success or failure of the course.

Summary

In the introduction of this paper a course was defined as an artifact; a structure or tool that enables the student to deal with an unknown exterior environment. Implied in this definition is the concept of good design; in other words, if the tool is designed in accord with the goals and purposes of the tool user it will be an *effective* interface. Further implied is the fact that if the tool is well designed then the basic principles of its construction can be known--such knowledge making possible tool replication and modification relative to the differences inherent in a new tool-using context.

The purpose of this paper is not one of course replication--academic environments differ too much for that. Rather our intent is a sharing of basic instructional design principles that we have identified as being important components in a general structure of any introductory course on computers and their applications in education (Tennyson, 1975). How the structure is filled in is really a matter of academic context, a matter that should be guided by the following principles:

1. Course objectives should be overlapping so that the student is put in a position of integrating concepts regardless of what objective he or she is working toward (C. Tennyson & R. Tennyson, 1976).
2. A course about computers in the classroom should expose its students to hands-on experience with a computer in a classroom setting (Fleming & Levie, 1978).
3. The classroom component of this type of course should contain a variety of learning activities--lectures, guest speakers, and films to name a few. The main factor here should be a consideration of the speakers' or materials' ability to deal with their topics in a clear and non-technical way (Popham & Baker, 1970).
4. Outside course readings should be closely integrated with all other class components and selected on the basis of its ability to deal with a technical topic in a way that is appropriate to the students' backgrounds (Klausmeier, 1976).
5. An end of quarter evaluation should be included as an effective monitor of students' perceptions (Tennyson, 1977).
6. All examinations should be of a mastery nature so as to promote subject learning rather than negative subject-related affect (Bloom, 1976).

There is one overriding principle inherent in all of the above, a principle that we believe forms the crux of the basic structure inherent in a course that, in our judgment, works effectively: It is a sensitivity and consideration, present during the design phase, that takes into account both the student's non-technical background and related feelings toward that lack of familiarity. Too often courses of this nature are designed without regard to this very important student predisposition--the result being courses that overshoot their mark; courses that overestimate student background, motivation, interest, and subject-related feelings. It is our belief that where such assumptions are present, anxiety will also be present because of undefined terms, poorly selected materials, and one-shot tests. Such a design flaw can only lead to more negative course-related affect.

If courses such as "Computers in the Classroom" are to be prerequisites to other and more in-depth courses in the field--introductions whose function it is to demyth the Medusian image of the computer by promoting positive experiences with this technology--then we cannot afford to leave students with negative course-related feelings. This point cannot be stressed enough if we are going to achieve true computer literacy in affective as well as content areas.

References

Apsell, P.S. (Producer). *The mind machines.* New York: NOVE/WNET, 1978. (Film)

Bloom, B. S. *Human characteristics and school learning.* New York: McGraw-Hill Book Company, 1976.

The computer age. Armonk, New York: IBM, 1977.

The computer society. *Time*, February 20, 1978, pp. 44-59.

Davidson, J. The square root of Tuesday. *Understanding media.* New York: Harcourt Brace Jovanovich, 1974.

Edwards, K. The electronic newspaper. *The Futurist*, 1978, 12(2), 79-84.

Fleming, M., & Levie, H. *Instructional message design.* Englewood Cliffs, N.J.: Educational Technology, 1978.

A glossary of terms. Unpublished course handout, 1978. (Available from Instructional Systems Lab, University of Minnesota, 178 Pillsbury Dr., S.E., Mpls., MN 55455).

Hardware at a glance. Unpublished manuscript, Northwest Regional Educational Laboratory, 1977.

Kay, A. C. Microelectronics and the personal computer. *Scientific American,* 1977, 237, 231-244.

Kemeny, J.G. *Man and the computer.* New York: Charles Schribner's and Sons, 1972.

Klausmeier, H. J. Instructional design and the teaching of concepts. In J. R. Levin & V. L. Allen (Eds.), *Cognitive learning in children.* New York: Academic Press, 1976.

Marks, S. The ubiquitous computer. *Understanding media.* New York: Harcourt Brace Jovanovich, 1974.

Merrill, M. D., & Tennyson, R. D. *Concept teaching: An instructional design guide.* Englewood Cliffs, N.J.: Educational Technology, 1977.

USE OF THE QUEST TEST MANAGEMENT SYSTEM IN A MULTI-CAMPUS ENVIRONMENT

Meg Lewis, Alan Villiers, Betty Rudnick
Texas Woman's University

The College of Nursing at Texas Woman's University has more than 2500 students at four widely-spread campuses. To ensure a consistent testing program and to simplify the paperwork involved in tracking students at multiple locations, the QUEST system (from the University of Texas Health Science Center at Dallas) has been adopted. Using QUEST, tests may be generated, administered, and graded. Test materials in psychology, medicine, pediatrics, nursing, obstetrics-gynecology and management are currently being developed and evaluated.

THE NEED AT TWU

The Texas Woman's University (TWU) is a medium-sized university having substantial student populations at four campuses in Denton, Dallas (two campuses) and Houston, Texas. Many of the instructional programs at TWU operate on more than one campus, with central direction in Denton but with resident faculty at the remote campuses. The undergraduate Nursing program provides particularly difficult problems in coordination: Nursing students typically take their first two years at the Denton campus then move to clinical work in Dallas or Houston; however, all students must take a common accreditation exam in their senior year.

To provide for consistent grading and to prevent redundancy of effort at the various campuses, the School of Nursing at TWU is now moving toward doing all testing from a centrally-maintained database of test questions. Test questions are being submitted from the faculty at all campuses, and screened through an editorial board; then they will be included in a set of question databases kept on-line on the central DECsystem-20 computer in Denton. The central computer is accessible to all campuses through either local or leased long-distance telephone lines; thus, access to these question databases is possible to all components of the Nursing program.

The effort involved in assembling, editing and entering the questions is quite substantial (perhaps 3000 questions will be entered in the next year). However, improved coordination of the Nursing testing program is being made possible by a flexible and elegant software package called QUEST, which was obtained from the University of Texas Health Science Center at Dallas. This package has programs to support the entry and editing of questions, the generation of tests and re-tests according to a number of criteria, and the maintenance of statistical indices about student performance. Once the initial editorial and secretarial effort is completed, the benefits of using QUEST will be substantial.

DESIGN OF THE QUESTION DATABASE

The QUEST system consists of a group of computer programs which allow one to create and maintain a database of multiple-choice questions. From the question database, the QUEST system can be directed to produce tests according to the user's specifications: tests may either be sent to a file for printing, or they may be administered directly at the computer terminal. In addition, there are programs in QUEST to edit the database, to do various maintenance functions on the database and to keep track of basic statistics.

Since the entire set of programs revolves around the question database, the best way to understand the power of the QUEST system is to know what goes into the database. For each question, there are several data items.

Each question MUST have:

a. Question number (assigned by QUEST)
b. Text (body) of the question
c. Correct answer
d. Question type

and a question MAY additionally have:

e. Categories (up to 4 of them)
f. Source code and source number
g. Optional response texts

h. Statistics indicating student performance on the question.
i. Access codes
j. Series pointers

For example:

```
Question #6  Answer=E  Qtype=0  Source=0-1
Edited=03-Jun-79  PBI=256  Diff=90  N=150
Percents = 0  0  90  0  10  AC's = 0  0  0
Category = ROOTS
***************
Which of the following numbers, when squared, is
EXACTLY 2.0.

a) 1.0
b) 4.0
c) 1.41421
d) 3.14159
e) None of the above.
***************
Response text B:
4.0 is the square of 2.  I'm looking for the
square ROOT.
-------------------
Response text C:
Ha!  I got you.  If you square 1.41421 you get
exactly 1.9999899241 which is not EXACTLY 2.0.
-------------------
```

b. The text of the question consists of one or more lines of text, in upper and lower case, stored exactly as the typist enters them when the question is added to the database. The text of the example question above consists of all lines between the rows of asterisks.

c. The correct answer to the question must be recorded when the question is entered (although it may be changed later.) In the example, the correct answer is 'E'.

d. In addition to the standard multiple choice, a few simple variations are available. This is the "Qtype" field. There are currently eight different Qtypes, including the standard 5-possibility multiple choice format, true/false questions, and a matching form. The other question types permit answers based on associations between items and on multiple correct answers: these types are included because of their common inclusion in the National Board exams in medicine.

e. A question database may contain up to 511 "categories", and a question may be included in as many as four categories. The categories subdivide the data base into various subject areas. A database in general mathematics, for example, might have categories such as "Algebra", "Geometry", "Complex Numbers", and so forth. Category is used as one kind of criterion for retrieving questions randomly from the database.

f. The "source code" and "source number" fields were originally designed to facilitate the sharing of data bases between institutions: each contributing institution would be assigned a source code. The source's question number could also be entered, since one source's number 7 need not be the same as our number 7. At TWU the source code is used to designate the campus which contributed the question to the database.

g. For the situation where the test is administered at the terminal, there are "default" messages which will appear in response to the student's answer: unless specified otherwise, the QUEST program replies with either "Sorry, that's not right", or "Correct!". An option exists, however, to include a more specific reply in response to a particular answer to a question, perhaps to tell the student WHY the answer is incorrect or to point to a relevant reference. In the example above, the computer will respond with;

'4.0 is the square of 2. I'm looking for the square ROOT.'

in the event that the student responds to the question with 'B'. Response texts, like the main text of the question, may be arbitrarily long, permitting the question designer considerable flexibility.

h. QUEST keeps a set of statistical indices associated with each question. These are:

1) Number of students attempting question.
2) Point Biserial Index (an index of the validity of the question).
3) Percent of students choosing A, B, C, D, and E (5 numbers).
4) Difficulty index (100 minus percent choosing right answer).

QUEST can update these statistics based on on-line student answers or via direct entry from other test grading packages. When generating a test, the user may select questions according to statistical criteria as well as category. For example, the user may restrict the questions chosen for a test to only those in a particular difficulty range and/or a particular point biserial index range.

i. In addition to the source codes, each question in a QUEST database may be associated with up to 3 numbers called "access codes". The codes may be used for any desired purpose, and may also be used as selection criteria when generating tests.

j. A particularly useful feature of the QUEST system is the ability to tie questions to a "series header" question. A "series header" consists of a block of text and a list of questions in that series. For example, a series header might read:

```
****************
A 20-year old male is brought to the emergency
room in coma, respiration 8/min, blood pressure
110/60, heart rate 80, rectal temperature subnor-
mal.
****************
```

Several questions might be in this series, concerned with different aspects of the situation presented by the header. In QUEST a series header is stored as a separate question - but with an indication that it is a series header. When a test is generated which includes a "series" question, the series header associated with the question is automatically included.

THE PROGRAMS IN THE QUEST SYSTEM

Three important tasks are handled by the programs in the QUEST system: a) database initialization and housekeeping, b) question entry and editing, and c) test generation. Intialization and housekeeping are done by interactive programs which: set up a new database, turn on (or off) optional features such as response texts, rebuild the database for more efficient storage after extensive editing, and provide listings of selected portions or all of a database. Question entry and editing are done through a program named QEDIT, which prompts the user to either add a new question or revise an existing question in the database. Both QEDIT and the housekeeping programs are typically used by a typist: these programs do not require any computer training beyond the ability to log on and follow simple directions provided, for the most part, by the programs themselves.

Unlike the other programs in the system, the program named QUEST - which is used to generate tests - may be used by a student for the purpose of testing himself or herself at the terminal. Questions may be retrieved either by question number, or may be selected randomly according to category type and statistical criteria. The person generating a test for off-line administration may also have the option of rejecting questions which are randomly selected by QUEST. If the questions are to be used for off-line testing, the question numbers selected are stored in a file which is given as input to another program named QTEST: it is the function of QTEST to produce a disk file containing the selected questions in nice form (sorted and arranged according to question type and category, for example) along with an answer key for the test. The test file can then be printed out and duplicated for distribution to the students who will take the test.

A WORD ABOUT SECURITY

When a general-purpose timesharing computer is used to store data for student testing, it is reasonable to be concerned about the security of the question database. Some measure of privacy is provided by the fact that all accounts on the DECsystem-20 are set up with individual "passwords": thus, the question databases may be at least superficially protected by careful distribution and frequent changing of passwords. Physical security is also provided at TWU by a daily "backup" procedure which copies all disk files accessed during a given day onto magnetic tape for off-line storage.

Coupled with these standard security measures, the QUEST system maintains its data files in compressed binary format, with question indices and question texts maintained separately. This storage mechanism both increases the operating efficiency of the programs which access the databases, and also provides another level of protection from prying eyes: it would take a student with considerable know-how and considerable time and access to the system to decode the question databases.

USE OF QUEST IN THE NURSING PROGRAM AT TWU

The Nursing curriculum at TWU is divided into roughly six subject areas, each of which is being entered into its own database: psychology, obstetrics-gynecology, medicine, pediatrics, nursing and management. The categories attached to the questions as they are entered indicate both a more detailed subject area and a particular course number in which the question is encountered. The source codes will be used to identify the campus from which the question was submitted (to give credit to the author).

Although at a later time the central databases will be used to generate ALL tests for the Nursing program, the first goal of the QUEST project is to produce the comprehensive exam which all senior Nursing students must take. Because of the lack of large numbers of terminals for student use, and to protect the privacy of the question databases, the tests will be generated for off-line administration. To ease the administrative burden, however, the tests will be answered using optical mark-sense forms. Grading and reporting of scores will be handled by existing test-grading programs available on the DECsystem-20.

One particularly nice feature of the QUEST system for an application such as this, where year-to-year evaluation of many students is a necessity, is the inclusion of statistical data in the question databases: this provides strong support for the continuing effort to validate and improve both the questions and the curriculum.

OTHER USES AND BENEFITS OF QUEST

Other users of QUEST at TWU have been the departments of Chemistry and Consumer Science. Like Nursing, the Chemistry department uses QUEST to generate off-line tests, in this case for the large freshman chemistry classes. For this application, QUEST provides a convenient way to generate printed tests and answer keys. The Consumer Science department, on the other hand, has used QUEST as a study aid: students were given free access to a database of questions concerning Housing, and were invited to study for their "real" test by browsing through the QUEST database. The student response to this use of the computer has been enthusiastically positive.

The QUEST system seems ideally suited to support both the large multi-campus (or multi-university) "consortium" of test questions as well as the on-line question browsing mode. In short, QUEST provides a convenient mechanism for maintaining and using databases containing any kinds of questions which can be stated in one of a variety of multiple-choice formats. Although an instructor could use the QUEST system with the optional response text feature to provide a rigorous computer-assisted drill and practice setting, perhaps the key benefit of QUEST is its complete and flexible handling of the administrative detail involved in creating tests. By putting the paperwork burden on the computer, the instructor can be free to put more emphasis on teaching, and the statistical data provided by QUEST may assist in analyzing the goodness of individual questions in the databases.

ACKNOWLEDGEMENTS

The QUEST software described in this paper was developed by and is available from the University of Texas Health Science Center at Dallas. The system is written in FORTRAN and MACRO-10, and it will run without modification on a DECsystem-10 or DECsystem-20 computer. For more information contact:

Steve Bush
Univ. of TX Health Science Center, Dallas
Medical Computing Resources Center
5323 Harry Hines Blvd.
Dallas, Texas 75235
Phone: (214) 688-3681

We are greatly indebted to Mr. Bush for his excellent manuals and courteous assistance while we were learning the system. We also wish to thank Mr. Mike Barnes, for his modifications of the statistics-gathering routines so that they would run on the DECsystem-20.

INSTRUCTION OF APPLIED MANAGEMENT SCIENCE TECHNIQUES IN A TWO-YEAR BUSINESS PROGRAMMING CURRICULUM

Don Mann, CDP
Department of Computer Sciences
Oakland Community College
Farmington Hills, MI 48018

Graduates of community college and other 2-year business programming curricula have occasional on-the-job need for a working knowledge of statistical, project control and optimization techniques, both at the request of the end user and in their own analytical and design work. A course offerred at Oakland Community College is described in which students use package programs to study and solve problems in the selected topics of central tendency, correlation, linear regression, critical path mathod, linear programming, and descrete deterministic and probabilistic simulation. The course thus parallels the professional environment in which the programmer/analyst would not code original programs but employ purchased or other previously written programs to perform analyses of this nature.

THE NEED

Programmers and analysts in the data processing centers of a wide variety of corporations have occasional need for a working knowledge of statistical, project control and optimization techniques. This requirement may come from regular or one-time computer services requested by user departments, or the need may occur in the analytical and design work of the D/P staff itself, as business systems are created for those departments. Many of these analysts and programmers are graduates of a two-year business programming curriculum such as those which are popularly offered by community colleges, but too few of these schools include education of this kind.

Further, in a typical job environment the programmer or analyst would not do the original coding to solve problems of this Management Science[1] nature, but make use of package programs purchased either from the hardware vendor or a software firm. Several such systems of programs are available. What the programmer/analyst thus needs in their educational background is basic knowledge of the specific O/R topics, experience in using package programs, and the ability to interpret for the end user the reasonability and validity of the results produced by the package programs.

The desirability of such content in a two-year curriculum has been addressed by the ACM Community College Committee in 1977 saying, among other recommendations, that a community college curriculum should include mathematical and statistical skills and vocabulary to better facilitate problem description, analysis and solution. A further recommendation was that the student should be able to utilize pre-programmed routines and utilities. Most models for two-year curricula based upon the ACM recommendations, however, do not devote an entire course to this area. As the profession of business analysis and programming advances, such knowledge is quickly becoming a necessity.

ONE SOLUTION

A course with the above features was proposed by the Computer Sciences faculty of Oakland Community College to the college's Data Processing Advisory Committee in 1972 and with their approval was incorporated into the curriculum. The course, Quantitative Computer Methods, is now a requirement in all three curriculum options offerred by the department, those options being Business Computer Programming, which is a two-year vocational option with the graduate directly employable as a programmer trainee, Computer Science, from which the graduate normally transfers to a four-year program at another school, and a Systems Analysis option, an evening program in which the student typically is already employed in data processing. A total of over 700 students are currently enrolled, with Business Computer Programming being the most popular option.

Since at present there is room for at most one Management Science course in a typical two-year business programming curriculum, only the subjects for which an analyst or programmer would have the greatest need can be included. The O.C.C. course incorporates the following topics:

I. Statistics (2 weeks)
    A. Single Variable Analysis
        1. Frequency Analysis
        2. Central Tendency
    B. Bi Variate Analysis

---

[1] For purposes of this paper, the terms Management Science, Operations Reserach and O/R are considered to be synonyms.

        1. Scattergrams
        2. Linear Regression
        3. Correlation

II. Critical Path Method (CPM) (1 week)

III. Linear Programming (2 weeks)
    A. Maximum Profit Systems
    B. Minimum Cost Systems

IV. Simulation
    A. Procedural Languages (2 weeks)
        1. Deterministic Simulation
        2. Stochastic Simulation
    B. Simulation Languages (7 weeks)
        1. Survey of Simulation Languages
        2. GPSS (General Purpose Simulation System) Language

Students attend three hours of lecture per week for fourteen weeks and are tested in all major topics. In addition, they solve problems from the textbook and perform a simulation project using programs available on the O.C.C. computer laboratory facilities, described later.

Please note that many important topics of Operations Research such as higher statistics and probability, Markov Chains, Transporation problems and nondiscrete system simulation cannot be included in such a one-term course. Extensions of some of the included topics, such as PERT-Cost, integer programming and nonlinear programming also cannot be covered.

COURSE DETAIL

The basic philosophy of the course is that the material must be computer-applied. A business programmer is not expected to be an O/R expert. A programmer can be expected, however, to know basic theory and be able to recognize O/R problems and apply package programs to solve those problems. Thus emphasis is also placed on methods of determining the reasonability of results from package programs.

In the statistics section, the students are instructed in the construction of frequency tables and histograms, including determination of number of classes, interval size, class boundaries and tallying. Central tendency topics include mean, median, mode, variance, standard deviation and skewness. Emphasis is placed on the use of these statistics with several examples being taken from business systems analytical work. Homework in these topics is required. Bivariate analysis addresses linear regression and correlation. In addition to basic theory interpretation of computer results to the end user is stressed. Students individually perform four bivariate laboratory problems taken from the text.

CPM (Critical Path Method) is a means of controlling large projects made up of many individual but related tasks such as found in the construction industry. The students learn how to compose a CPM network, solve small projects by hand, and perform two computer solutions on problems supplied by the instructor. PERT (Program Evatuation and Review Technique, an historical variant of CPM) and PERT-Cost are not covered.

Linear Programming is a method of distributing limited raw resources in the fabrication of finished products to achieve maximum overall profit. It can also be used to realize minimum overall cost among products the company must purchase. In addition to seven laboratory problems taken from the text, the students use the computer system to determine their own individual diet from among foods they like while meeting nutritional needs, the "cost" being calories. (Some newspaper advertisements offer this same service, at a not inconsiderable sum)

Approximately the last half of the course is devoted to discussion of simulation of discrete systems, in which transactions such as invoices or products move through a system such as an accounting office or a manufacturing process, contending with each other for the service of various facilities.

Attention is first given to how this can be accomplished in procedural languages such as FORTRAN, BASIC, APL and others. Central to this discussion is the concept of random number generators available as functions in most languages. It is shown how to construct your own if the language does not supply one. Although all students will have had at least one programming language, their programming backgrounds vary. Board work discussion at this point is therefore accomplished via flowchart presentations. Just for fancy, an age-structured model of the Michigan turkey (sic) population, written in Fortran, is included among the examples.

After a survey of the various popular languages that are specifically designed for simulation such as GPSS, CSMP and SIMCRIPT, the remainer of the course is devoted to learning the fundamental commands of the General Purpose Simulation System (GPSS) language. For lab work the students divide into 3 or 4- person teams and develop team projects. Each team selects a system to model, defines what is to be optimized (e.g., hospital patient time, bank teller productivity), creates a working model of the present real-world system in GPSS, adds enhancements aimed at optimization, and then reports their findings in a formal presentation at the end of the term to the rest of the class.

Student simulation projects have ranged from the macroscopic, such as the highway patterns at the Pontiac Silverdome in which the traffic snarls currently experienced by that facility were individually predicted with rather amazing accuracy, to the very detailed simulation of a simple four-way traffic intersection in which each square foot of the intersection was probabilistically allocated during five cycles of the traffic light.

Some of the more popular models have been the college registration area, the college cafeteria, the computer laboratory, banks, fast-food restaurant, accounting departments and hospitals emergency wards. One innovative team simulated the traffic patterns in a men's restroom, collecting live data in time-and-motion-study fashion by an observer complete with clipboard. He escaped unharmed. The recommendations of another team were actually incorporated by the manager of the fast-food restaurant they observed with the results being very positive, but only to the degree predicted by the study.

LABORATORY WORK

The instructional computer laboratory at O.C.C. currently houses a PRIME 400 computer system, with 1.5 megabytes of real memory, 104 megabytes of disk storage, 37 student CRT terminals and a leased line to the administrative computer system of the college, an IBM 370/145, located twenty miles away. The instructional computer laboratory is a "punched-card-less" facility. Via the CRT terminals the 700+ students maintain programs and data sets in their individual libraries kept on disk. They are allocated a certain number of connect-minutes per week by the system for each course in which they are enrolled. Other resources such as CPU seconds, local prints and priority runs are also maintained. Although they may accumulate a surplus of any individual resource from week to week, once they deplete a resource they must wait until the following Monday when resources are again distributed. The system will refuse to acknowledge them at a terminal if their connect time or CPU time has a zero balance.[2] Both the hardware and the software developed by Ron Pavlak, a member of the faculty, place this among the most sophisticated community college computer facilities in Michigan.

The Quantitative Computer Methods students interact with preestablished programs on the PRIME system to complete their labwork in statistics, CPM and linear programming. They use the system to compose GPSS simulation models and have the system transmit those to the IBM system for processing, with the print-out being returned to the Orchard Ridge Campus in Farmington Hills.

THE TEXTBOOK DILEMMA

There is no good textbook for this course. Texts which include all the topics are much too theory-bound for a programming student and are not computer-applied. Others containing O/R computer applications do not include all the above subjects. A middle-ground text containing the more popular O/R topics in which the theory is addressed but the emphasis is on how to apply the concepts via package programs, and on the interpretation of computer results is needed. The present text[3] was chosen in 1976 because it is very well computer-applied, it is well written in those O/R topics it does cover, the topics it includes which are not in the course may be useful in other courses, the exercises are numerous and meaningful and it is a softback and thus less expensive to the student. The topics of CPM and GPSS must be covered by handout materials, since they are not included in the text.

PLANNED ENHANCEMENTS

To further provide applicability of the course, arrangements are being pursued with the Business Administration faculty so that the O.C.C. Management course and this course may be offered concurrently, with the students forming joint management-programmer teams to solve some optimization and simulation labwork, similar to an actual work experience.

Although the package programs used in this course were developed by the author, a survey is being made to determine if a particular statistics package is significantly more popular among the corporations in the service area of the college. If so, it will be purchased and incorporated into the course labwork.

Finally, unless informed of some other recent work, this author is pursuing the development of a text which will cover simple statistics, CPM, linear programming and GPSS Simulation in a computer-applied framework, with heavy emphasis on package program solution to a large collection of included laboratory problems.

REFERENCES

1. Little, J.C., "Computer Education and Community Colleges", Interface, The Computer Education Quarterly, Vol I, Issue 1, Winter 1979, pg. 12.

2. Oakland Community College, O.C.T.O.P.U.S., Terminal Users Guide Login Procedure and System Commands, Fall 1979.

3. Barrodale I., Roberts, F.D.J., and Ehle, B.L., Elementary Computer Applications, Wiley & Sons, N.Y., 1971.

COMPREHENSIVE SCHOOL FACILITIES INVENTORY SYSTEM

PHILLIP P. MARINO

MARYLAND STATE DEPARTMENT OF EDUCATION

111

Introduction:

The purpose of this paper is to present an overview of the educational administrative, and technical requirements that initiated the specifications and implementation of the "Computerized School Facilities Inventory System" for the Maryland State Department of Education. The Facilities Inventory System is completely operational and utilizes an IBM 370-158 and VSII, however, it has been written in COBOL and designed to be autonomous of any one vendor computer so that other agencies may utilize the programs with minimal amount of changes for their own computer system. It was designed, at its inception, to be multidemensional and integrate the information requirement of educators, legislators, administrators, planners and the community.

Background:

Before I begin the discussion on the facilities inventory, you need to know the background concerning the Agency that initiated the request for the facilities inventory. The Interagency Committee for Public School Construction (IAC) was established by the 1971 session of the Maryland State Legislature. This agency is responsible for the administration of a capital improvements program which is expected to be approximately $1 billion, 325 million by fiscal year 1981. It is also responsible for the coordinating activities of employees of the Department of Education, the Department of General Services and the Planning Department. This Agency also has the accounting responsibility for all financial payments, auditing and projecting bond sale requirements, and due to its multipurpose concept is continually answering questions not only from the local subdivisions but also from educators, legislators and concerned individuals.

On November 7, 1974 the Board of Public Works approved funding to provide a public school Facilities Inventory System. The purpose of the Facilities Inventory System was to compile and process architectural, structural, mechanical, cost and room space data concerning all public schools throughout the State. This inventory system would allow the Interagency Committee a better assessment of construction and facility needs while determining funding priorities. The main function of the Facilities Inventory System is to assist the IAC in assessing construction needs and controlling construction funds from loacl subdivisions.

General Requirements:

Specifically, the Facilities Inventory System (FIS) was designed to support the following functions:

1. To establish an information bank, to assist in determining facility needs and in obtaining/promoting new ideas concerning educational facilities.

   This information bank includes a statewide inventory of educational facilities from which standards, averages, medians, variances, etc.'s of various facility characteristics can be derived for selected levels/groups of schools (county, state, program, student, age, demographic parameters and so on).

2. To establish for the local level a guide to the process of developing an explicit statement of facility needs and funding requests and to support this process with a future school modeling service.

3. To establish a means for deriving funding priorities based on examination of needs and characteristics of current facilities as reflected by the information bank.

4. To provide a measuring stick in terms of composite profiles of current and/or proposed facilities, thus allowing for the further evaluation of construction requests.

5. To support the IAC and the Maryland State Department of Education's reporting requirements to other governmental agencies by automating as much of the reports as possible.

6. To provide information for legislators and other concerned individuals in the community.

Initial Problems:

In 1972 outside contractors were requested by the IAC to produce a Facilities Inventory System and within a year had developed a series of computer programs that generated a series of computer reports. However, many of the data elements that were identified in the original request were much too difficult, if not impossible to secure; and consequently upon completion of this system, the generated computer reports were unusable. Additionally, during the programming period, many of the requirements had changed and rendered many of the original data elements obsolete. Another critical area which did not receive much attention at the time was the constant need for ad hoc reports. It was a combination of problems that caused the system to be less than what was anticipated. The first problem was that the user did not know its needs and requirements. The second problem was that the consultants did not take the time to clarify the user's needs. At that point, frustration set in at the IAC, and they realized that someone with data processing expertise was needed on their side who could understand their needs and requirements and convey those needs and requirements into a computer system that would give them what they need. The Interagency Committee then contracted with the Maryland State Department of Education for a systems consultant, who would be assigned to them and for them.

I was assigned by the Department to implement the Facilities Inventory System as designed by the contractor. My recommendation based on the analysis of the system, its function and what the user required, was to discontinue any further work on the system. My analysis demonstrated that over one-third of the existing data elements were obsolete and, during the programming interim, there was a twenty percent increase in new data elements. My analysis also considered the fact that the system as designed did not address the critical ad hoc reporting functions which were crucial to this Agency's purpose.

My recommendation therefore was that the entire system be redesigned to function as a quasi-data base system and would be handled in two phases. The first phase would be to edit and update all data elements collected to create a master file. The second phase would be to utilize the flexibility of a data management system (generalized computer software package), to extract appropriate information that is required by the user. The advantage of a generalized computer software package is that the package is specifically designed for the purpose of selecting, retrieving, manipulating, calculating and report generation of any data currently residing on the user's file. Additionally, this software package does not have to be programmed (as it is a prepackaged commodity available from various vendors) and is specifically designed to resolve the user's ad hoc reporting requirement.

System Development:

The main concept of the Facilities Inventory System is that each school building is a record unto itself and that in this record there is information regarding not only the school number, school name, address, etc., but there is additional information identifying architectural, mechanical and electrical subsystems within that building. There is also additional space within the record that contains information regarding categories of space allocation and space utilization. Let me clarify this point. Planning personnel, in conjunction with Educational and General Services personnel, felt that there were two major categories of space, Programmed and Unprogrammed; and within the Programmed space there were eight major categories, namely, Instruction, Research and Study, Physical Education, Assembly, Students, Teachers, Administrative and Food Service. Within the Unprogrammed space were Utility Support and Circulation. These categories become important later on since the budget allocation from the local subdivision include information regarding the allocation of space regarding Programmed versus Unprogrammed space.

In the computer, the detailed information collected concerning each room/space is consolidated and accumulated and placed in the appropriate major categories on the master record. Further, the detail that created these categories are placed on a separate file for later access for special ad hoc reports utilizing the generalized computer software package. In addition, and as part of the school master record, costs are collected by 16 major categories utilizing American Institute of Architects (AIA) nomenclature starting from General Conditions, Sitework, Concrete, etc. through Electrical Systems. Within these 16 major categories, 11 subcategories are also collected specifically identifying costs allocated to Foundations, Structures, Exterior Clading, HVAC (heating, ventilation, air conditioning) through Lighting. Last, but not least, in the design of this record and knowing the continuing changing educational environment, sufficient space was allocated as part of the record for future expansion.

Data Collection Procedure:

Another very critical area of design not addressed in the original system design but addressed here, was the aspect of collecting data. Procedures have to be developed that identify not only the type of data collected but who, when, where, and how the data would be collected and inputed into the system. My recommendation that the Interagency Committee standardize all requisition forms and make these forms mandatory for all contractors. This Standardized Requisition Form incorporated the Construction Cost Index Breakdown provided by the American Institute of Architects and utilizing these cost breakdowns created a source document that was then made part of the contract documents, thereby eliminating the major problem of data collection. The Facilities Inventory System is fully functional and operational and is currently producing an annual inventory control report listing each school and every data element available concerning that school. An ancilliary report produced in this system is a cost analysis report which enables not only modeling capabilities in terms of composite profiles of current facilities thus allowing for further evaluation of construction requests but also enables cost overruns to be highlighted by displaying not only the total cost per square foot of the building but also the individual costs per square foot of each of the 16 major categories and each of the 11 minor categories of costs previously identified.

The ad hoc reporting phase of the system is also fully operational and functional and, in fact, is not only being utilized by the Interagency Committee but by many different departments and agencies within the State for their own data management requirements. Examples of the ad hoc requests given to this Agency are reports that reports schools by capacity and their cost per square foot by school types, a report that identifies buildings that are 8 years or older, 20 years or older, 30 years or older, etc., a report that contains information regarding schools that have electrical kitchens or utilize oil heat, a report that lists the total gross square footage for all buildings in the State of Maryland, and the reports go on. In the majority of these cases, the reports were available by the next day, which clearly attests to the success of utilizing the data management concept.

The only major problem we are experiencing with this system is that comprehensive data has not been collected and verified for all those schools subsequent to the standard requisition, and a concentrated effort will have to be made by the Interagency Committee to secure this data.

The Future for the Facilities Inventory System:

Where does the Facilities Inventory System go from here? As the Facilities Inventory System is used, it will gain experience and be refined. Testing and evaluation of this system will continue to improve its reliability and credibility, but even though the Facilities Inventory System is a new system, it is extremely useful at the present time.

The future of the Facilities Inventory System is very exciting. We have utilized data base technologies in the design of the Facilities Inventory System, and we fully expect to provide access to this System to appropriate local subdivisions (LEA's) via terminals at their locations to meet their informational needs and requirements. We will also be able to provide access to this information base at legislative hearings via portable terminals.

The potential is enormous and the possibilities exciting. Both can only benefit the community and the State.

COMPUTING IN MINORITY INSTITUTIONS: AN UPDATE

Sister Patricia Marshall
Xavier University of Louisiana
New Orleans, LA 70125

Donald L. Alderman
Educational Testing Service
Princeton, NJ 08540

Richard Jaeger, School of Education
Univ. of North Carolina-Greensboro
Greensboro, NC 27412

This paper updates and extends a report on a minority institution subset from the Fourth Inventory of Computers in Higher Education (1976-1977). The update is based on additional analysis, acquisition of supplementary data, and a revised sample of minority institutions. There are implications for all institutions, as well as minority institutions.

Introduction

When data were gathered in the Fourth Inventory of Computers in Higher Education[1] (hereafter referred to as FICHE), a chapter was written for later publication on "Computing in Minority Institutions: 1976-77."[2] A study still in progress (and referred to in this paper as the ECMI ASSESSMENT)[3] has provided additional data on computing in the minority institutions from a questionnaire survey,[4] as well as further analysis of a slightly different version of the original sample.[5]

The questionnaire survey sample constituted a larger group of minority institutions with data from 1977-1978 and projections through 1981-1982. This study will be referred to in this paper as the QUESTIONNAIRE.

The FICHE sample of minority institutions, slightly revised, was compared with the total FICHE sample of all institutions. That study will be referred to in this paper as the FICHE UPDATE.

FICHE UPDATE

Alderman states in the introduction to his report that "the focus here is on access to computing resources at minority institutions and on applications of these resources...contrasts will be drawn between minority colleges and universities and non-minority colleges and universities."

"A college or university was designated as a minority institution if a majority of its student enrollment represented the following racial and national origin groups: Alaskan Indian, American Indian, Black, Eskimo and Hispanic. Each of these groups has a lower percentage of persons in the sciences than has the general population, and in this sense these groups may be viewed as under-represented in the sciences. All other colleges and universities, whether none or half of their students came from these same groups, were designated as non-minority institutions. Thus, the contrasts between the two types of institutions reflect an artificial dichotomy subject to change as enrollment patterns at particular institutions change. And the classification of an institution based on its student body would not necessarily be the same if classification instead depended on minority representation among its faculty."

Alderman's sample contained 202 minority institutions, of which 105 had completed at least one form in the FICHE survey, for a 52% response rate. He cautions that no independent confirmation has been made of the similarity between respondents and non-respondents, either in the minority sample or the non-minority sample (whose response rate was 59%). Nevertheless Alderman states that notable imbalances do exist with regard to enrollments and degree programs.

"Larger enrollments and higher degree programs" he states "often mean greater awareness and more widespread use of computing resources. Since a larger number and a larger proportion of non-minority institutions came from these categories, analyses should show minority institutions to be at a disadvantage with respect to computer access and applications. This expectation receives additional weight when the above-average response rates from non-minority institutions (i.e., a 65% response rate from non-minority institutions with 10,000-19,999 students and a 71% response rate from non-minority institutions with over 19,999 students) and doctoral degree non-minority institutions (i.e., a 67% response rate) are taken into consideration. Indeed, the non-minority institutions with large enrollments tend to be those which offer the doctorate degree."

Alderman's major findings in the FICHE UPDATE are summarized below:

"Minority colleges and universities offer only a small number of degree programs in computer science and related fields and, therefore, award a very low number of such degrees each year. The disparity between minority and non-minority institutions in this respect far exceeds their proportional representation in the population. Especially at the baccalaureate and master's degree levels there is a need to initiate and to expand degree programs in computer science and related fields at minority institutions.

"Seventy percent of minority colleges and universities had access to computing resources and sixty-eight percent of non-minority colleges and universities had access to computing resources. Despite their smaller student enrollments and lower degree programs, minority institutions have computers to the same extent that non-minority institutions do.

"Computer installations dedicated to specific applications in administration, instruction, and research show much the same pattern of computer use in both minority and non-minority institutions. There is also a similar pattern in the frequencies with which minority and non-minority institutions offer particular programming languages. Furthermore, comparable percentages of minority and non-minority institutions support remote modes of computing and interactive computing.

"It would appear that students at minority institutions do not receive as much exposure to computers in their academic studies as do students at non-minority institutions. Although differences in student enrollments and in degree programs account for some of the disparity in the total numbers of students using computers in their courses, minority colleges and universities reported only one-fortieth the total number of students with exposure to computers in academic courses reported by non-minority colleges and universities.

"The 105 minority institutions responding to the survey reported a total of 35 full-time faculty members with doctorates in computer science or related degree programs; the 1,707 non-minority institutions reported nearly 1,800 such faculty members. The under-representation of certain minorities in the computer-related professions may be attributed, in part, to the scarcity of appropriate degree programs at minority institutions; and the scarcity of such degree programs may, in turn, be due to a lack of key faculty members.

"Small baccalaureate minority colleges, those with an enrollment of 500-2,499 students, spent more on their computer installations than did comparable non-minority institutions. The greater average expenditure of these minority institutions arose primarily from capital costs for computer hardware and from operating costs from software services. These cost categories would be consistent with acquisition of computer equipment and with expansion of support services, perhaps indicative of recent entry into the computer field.

"These findings suggest that the initiation and expansion of degree programs in computer science and in related fields receive the highest priority for attention at minority colleges and universities," states Alderman in his recommendations. "Concomitant with this attention to curriculum programs should come concerted efforts to recruit faculty members in these disciplines. The under-representation of certain minorities in the computer professions seems less a problem of access to computing resources than of access to relevant degree programs and faculty members."

Alderman recognizes in his report that it cannot be inferred from the apparent parity in hardware between the minority and non-minority samples that the minority institutions have caught up with the majority in computing. "The nature of computer uses," he says, "whether administrative, instructional, research or some combination, reflects the manner in which academic institutions harness computer capabilities; and patterns of computer applications in specific courses can reveal particular weaknesses or strengths at minority institutions...Apparently the computing resources for academic programs in computer science and related fields exist at minority institutions. Other reports[4] may reveal whether the quality of these resources can support academic programs."

117

Alderman notes further that "the consistent pattern of computer uses for minority and non-minority institutions seems to contradict differences already observed in degree programs for computer science and related fields. From the very low numbers of such degree programs at minority colleges and universities it might be expected that there would be either a lower access rate to computing resources or a different pattern of usage at minority institutions than at non-minority institutions. Since the access rates for the two types of institutions were comparable (i.e., 70% at minority institutions and 68% at non-minority institutions), there should be differences evident in the patterns of computer use. There was no obvious difference in these patterns. Two explanations seem plausible: there may be differences in the quality and power of computing resources at minority and non-minority institutions not apparent from the quantitative report of access to computers, or the category including administrative and instructional or research uses of computers may obscure real differences in the allocation of computing resources to each kind of application."

With regard to Alderman's finding that non-minority institutions' students make much heavier use of computers than those in minority institutions, he staes that "clearly the large enrollments found at some non-minority colleges and universities must contribute to this vast disparity, but it is doubtful that size alone accounts for the difference. Facilitation of student exposure to computers at minority colleges and universities seems an appropriate response to this inequity."

Alderman also notes the exceptions to the general pattern of programming languages and certain modes of access among the minority institutions with regard to graphics capabilities and the PASCAL language, "which in turn imply a possible need for higher-level and more diverse support packages at minority institutions." He goes on to suggest that "such support packages would seem a derivative benefit if there were to be an increase in computer science programs at minority institutions accompanied by upgraded computer equipment and computing capabilities."

Alderman hints at one universal standard, to which he sees both minority and non-minority institutions not measuring up: "The fact that roughly two-fifths of the computer installations at both types of institutions had interactive computing available for supporting work on computers suggests that all colleges and universities should seek to increase accessibility to their computing resources."

On the other hand, Alderman, pointing to a serious imbalance in computer science degree programs, states that "the disparity in the number of responding institutions and the number of non-minority institutions with large student enrollments explain some of the sheer differences, these factors do not fully account for the gross imbalances with respect to faculty."

He also says that "the higher costs of computer installations" at the small baccalaureate minority colleges "probably reflects recent entry into the computer field rather than a higher level of sustained fiscal support."

Alderman concludes that "the problem of under-representation of minority groups in the computer professions appears not to be one of hardware or computing resources but of persons." He recommends that "if the imbalance of minority representation in employment positions in the computer field is to be addressed through concerted attention, that attention should be focused on relevant curriculum programs and faculty members at minority colleges and universities."

QUESTIONNAIRE

Some additional insight can be gained by examining the findings of Richard Jaeger, who performed exhaustive analysis on a large, complex data base gathered by means of a QUESTIONNAIRE. Different forms went to four groups: presidents or chancellors of minority institutions, academic deans or vice presidents, department heads, and academic computing directors (or their closest counterparts). These groups of respondents will be referred to in this paper as presidents, deans, department heads, and computing directors.

The QUESTIONNAIRE forms went to 239 minority institutions, of which 193 were in the FICHE UPDATE sample of 202. (About half the increase is accounted for by the addition of relatively new institutions, including some two-year Indian schools struggling against bureaucratic odds to become stable.) QUESTIONNAIRE data were from 1977-1978 for the most part, with some projections to 1981-1982.

Of the 239 institutions, 154 sent usable responses from one or more respondents, for an overall institutional response rate of 64.44%. Of the presidents, 96 (40.17%) responded. The 83 deans responding constituted a response rate of 34.73%, and 56 computing directors' rate was 23.43%. Four forms per institution were mailed for department heads. Some of the smaller institutions mustered only one response from this category. One large Puerto Rico institution copied the forms and sent back nine completed ones. Thus, a response rate for this category would have no meaning.

Excerpts from highlights which Jaeger listed (from the nearly 700-page, 500+ table report) follow:

"Responding institutions varied widely in the number of science faculty they employed: the smallest institutions had none full time and the largest had almost 500; the smallest fourth of the institutions had no more than 35 full-time science faculty and the largest fourth had at least 160 full-time science faculty.

"Presidents at 24 percent of the responding institutions stated that sciences are emphasized more than arts in their schools; 17 percent reported that arts are emphasized more than sciences; 57 percent reported equal emphasis.

"Responding institutions emphasized undergraduate education overwhelmingly: almost 90 percent of responding presidents reported this emphasis.

"Every responding institution reported enrollment of at least 80 percent minority students. Nearly sixty percent of the responding institutions reported a majority black enrollment, and over 40 percent reported at least 90 percent black student enrollment.

"Just over half of the responding computer center directors[6] in institutions having access to a computer reported the exclusive use of central computers; nearly ten percent reported the exclusive use of personal computers; almost 40 percent reported the use of both.

"With few exceptions, responding computer center directors[6] reported the availability of very few input/output devices for academic computing use at their institutions.

"Almost half of the responding computer center directors[6] reported that their institutions spent less than $10,000 on computing hardware during the 1977-78 academic year...88 percent...spent less than $50,000...

"About 2/3 of responding institutions[6] were reported to have spent less than $1,000 on computing software during the 1977-78 academic year; about the same proportion spent less than $10,000 to support computing personnel that year.

"Median expenditures for computing hardware during the five-year period 1973 to 1978 were in the $100,000 to $250,000 range at responding minority institutions.[6] Corresponding expenditures for computing software were in the $5,000 to $10,000 range for the five-year period.

"Forty percent of responding computer center directors[6] reported receiving some computing support from the federal government; support funds from their own institutions were cited by a greater percentage. Of the institutions that reported receiving any federal funds during the 1977-78 academic year, over 2/3 received less than $20,000.

"Nearly 90 percent of responding presidents ...agreed with the statement that 'Many students would (or do) benefit from a computer science program at this institution.'"

Most of the "responding computing center directors...wanted some combination of batch, interactive and personal computing for their institutions by 1981-82.

"Almost 2/3 of responding...deans reported that campus-wide groups had met at their institutions, to study the acquisition or improvement of computer facilities and capabilities for instructional purposes. Sixty percent of responding department heads reported that study groups had met for similar purposes.

"Almost 2/3 of responding presidents...reported that their institutions had a long-range plan for improvement of academic computing services.

"Only half of the responding computer center directors[6] reported that their institutions owned a computer used for academic purposes."

Problems in procuring or improving academic computing resources were seen by Jaeger's respondents as generally related to budget limitation and/or political problems at the local, state, and federal levels. "No president...reported specifically that lack of federal funds presented development problems, but rather that problems stemmed from restrictions on the use of funds."

About half the responding computer center directors[6] felt that institutional budget priorities were either a major or an extremely severe problem. Lack of expertise among both computing personnel and users was also seen as affecting the development of academic computing.

"Analyses of relationships between the size and scope of academic science programs and academic computing capabilities in minority institutions provided clear evidence of many linkages between these factors. For example, minority institutions were far more likely to have made efforts to improve their academic computing capabilities if they employed a large number of faculty members in the sciences."

Conclusions

Both Alderman and Jaeger present evidence to support the conclusion that although approximately the same percent of minority and non-minority institutions appear to have computer hardware or access to it, the use by students in academic courses is far lower in minority than non-minority institutions. Alderman especially points to a huge disparity between minority and non-minority institutions with respect to computer science degree programs which far exceeds their proportional representation in the population. Late entry into the computer field is cited by Alderman as a possible partial cause of the lack of degrees in computer science being offered by the minority institutions. Lack of interest in computer science and/or academic computing does not appear to be an important factor, as nearly all of the Jaeger respondents suggest that academic computing should be a large part of their institutions' activities by 1981-82, and nearly two-thirds of the institutions responding to his survey were reported to have convened campus-wide planning groups for study of the acquisition or improvement of instructional computing.

Alderman suggests that the approximate two-fifths of all institutions in his sample--minority and non-minority--that had interactive computing available for supporting work on computers are below a universal standard in that regard.

Size of institution and size of science faculty were found to have a bearing on the extent to which an institution successfully developed academic computing, according to Alderman and Jaeger respectively.

Alderman sees the higher computing costs at the small bacallaureate minority colleges as reflecting recent entry into the computer field rather than a higher level of sustained fiscal support. Jaeger's statements on spending in 1977-78 seem to support this contention.

Lack of computer science staff found by Alderman to be a major factor in the non-production of computer science degrees by minority institutions finds support in Jaeger's statement that about 2/3 of the responding institutions spent less than $10,000 to support computing personnel in 1977-78.

Recommendations

Alderman recommends, in view of his findings on the lack of computer science degrees being produced by minority institutions, that attention be given to finding ways of initiating and expanding computer science degree programs and faculty in the minority institutions, with the greatest emphasis on faculty. "The problem of under-representation of minority groups in the computer professions appears not to be one of hardware or computing resources but of persons," he says.

Jaeger, on the other hand, shows concern about the quality of the hardware available to students, finding few input/output units accessible to them. Together with the evidence that comparatively little of the use of computers in minority institutions to date is academic, this would seem to suggest that the mere fact that a campus has computing hardware or access to it does not mean that the hardware is suitable for academic use. The need for interactive computing (which Alderman sees as a universal need) seems even greater for the minority institutions, given the foregoing discussion.

It seems clear that the minority institutions have made a beginning, which is expressed in terms of hardware and plans. Planning should continue, and proposals should be made to improve and expand computer science programs and faculty, as well as academic computing generally. Moreover, it seems clear that funding agencies should give serious attention to the problem of computer science degree production in the minority institutions and to the problem of increasing interactive computing capability in all institutions.

REFERENCES AND FOOTNOTES

[1] Hamblen, John W. and Baird, Thomas B. (Eds.). Fourth Inventory of Computers in U. S. Higher Education 1976-77. Princeton, NJ: EDUCOM, 1979.

[2] Marshall, Patricia; Hamblen, John W.; Jones, Clinton; and Lewis, Jesse C. "Computing in Minority Institutions: 1976-77" (Volume II of Reference 1 above, Chapter 7, Pages 151-166).

[3] Marshall, Sister Patricia. ECMI Needs/Strategy Assessment (Educational Computing in Minority Institutions). Work in progress under National Science Foundation Grant No. SPI-78-21515. New Orleans, LA: Xavier University of Louisiana. To be completed July 1980.

[4] Jaeger, Richard M. Academic Computing in Minority Colleges and Universities. Greensboro, NC: University of North Carolina at Greensboro, Center for Educational Research and Evaluation, 1979.

[5] Alderman, Donald L. Access to Computing Resources at Minority Colleges and Universities. Princeton, NJ: Educational Testing Service, 1980.

[6] The response rate for academic computing directors in the questionnaire survey was not quite one-fourth. Follow-up telephone calls suggest that many were too burdened to respond, and many do not exist (except sometimes for a full-time professor who also supervises equipment). Caution should be exercised in interpreting the responses of this category.

COMPUTER LITERACY: A WORKSHOP APPROACH FOR HIGH SCHOOL MATH TEACHERS

Sister Joseph K. McAdams
Immaculata College
Immaculata, PA 19345

Immaculata College, a liberal arts college for women, is located within the Archdiocese of Philadelphia. The Archdiocese runs 30 high schools which take in grades 9 through 12. Recently, the Secondary School Mathematics Committee of the Archdiocese decided to include computer concepts in the ninth grade mathematics curriculum. The topics to be covered include the history of computer science and concepts of computer operation, as well as elementary programming in BASIC.

Although most of the secondary school mathematics teachers in the Archdiocese of Philadelphia hold at least a Master's degree, many received their training before the advent of computer science. Many younger teachers, through no fault of their own, were never exposed to computer science. It is only recently that a serious effort has been made to include computer concepts as part of the standard curriculum for a Bachelor's or Master's degree in Mathematics Education. Therefore, Philadelphia's decision to put computer science into the ninth grade mathematics curriculum caught many teachers unprepared. To assist some of these teachers, Immaculata College offered a two-week workshop during August of 1979.

Because the actual time of the workshop would be so short, the participants received a copy of the textbook and workbook in June, with instructions to read the text and to do as many of the exercises as possible. This was to allow discussion to proceed with a minimum of delay during the actual workshop. The text was a general data processing text that adequately covered the terminology and equipment, as well as a small part of the history of computing. Immaculata also arranged to borrow a Radio Shack TRS-80 Level II to supplement its own DEC VT/78 system. This would allow 3 persons to use a computer at a given time - 2 on the DEC and 1 on the TRS-80. Due to these limitations of equipment, it was decided to limit the number of participants to eight. Of the eight teachers chosen for the workshop, six already held a Master's degree and currently were teaching on the secondary level. The other two teachers were working on the upper level (grades 7 and 8), and were preparing to teach on the secondary level. Enthusiasm was high, so there were no problems of motivation. Activities were scheduled each day from 9 to 11:30 A.M., 1 to 4:30 P.M., and 7 to 9 P.M. Actual classroom sessions were held from 9 to 10 A.M., 10:30 to 11:30 A.M., 1 to 2 P.M. and 2:30 to 3:30 P.M.

BASIC Programming was introduced the first day so that the computer could be used immediately. Making use of two different computers for programming made the participants painfully aware of the fact that different versions of BASIC exist, and that these different versions are not transportable. All time not scheduled for class sessions was spent in writing and/or executing programs on the computer. It was not at all unusual to find all eight participants in the computer room working on the computers in groups.

Interspersed with the lessons in BASIC and programming techniques were the following topics:

History of Computer Science
Data Processing and Electronic Data Processing
Representation of Data
Data Recording Media
Storage Devices
The Central Processing Unit
I/O Operations and Devices
Computer Operations
Programming Languages
Files and Data Bases
Operating Systems
Data Communications
Computer Assisted Instruction

Particicipants had a choice of taking the workshop for 3 credits or simply for a certificate of participation. Three took it for credit; five for the certificate. Those taking it for credit were expected to fulfill the requirements of the normal Computer Literacy course offered at Immaculata.

This included the satisfactory completion of three examinations, proof of having read numerous articles from Computer World and other computer-oriented publications, completion of at least three programs in BASIC (all did more than three!), and participation in at least one field trip.

Two field trips were scheduled (all eight participated in both): one to visit PLATO at the local Control Data Learning Center, and one to visit the data processing center of the Archdiocese of Philadelphia. These field trips were scheduled for the second week so that the participants could appreciate what they were seeing. PLATO was a big hit. Enough discussion of computer assisted instruction had taken place to enable them to be properly impressed with PLATO's capabilities. The visit to the data processing center -- in the midst of running class schedules for the high schools -- gave them a good look at the various types of data processing equipment, as well as a chance to observe an on-line data base system in action.

All of the participants considered the two weeks to be very worthwhile. Their attitude was evident in the amount of time spent working with the computers. Programs were interchanged and translated into both versions of BASIC. Those with access (or hopes of access) to a TRS-80 made tapes of the various programs written by the members of the group. A hard copy listing was made of the programs written on the DEC machine.

The long-range benefits of the workshop are yet to be seen, but several short-term results are evident. Mainly, an interest in computing is being created among six different high school faculties. Some schools that are currently time-sharing are planning to purchase a computer. In at least one case, the decision to purchase a computer is a direct result of the teacher's participation in the workshop. Most important of all, there are a few more "computer literate" teachers on the educational scene.

COMPUTER APPLICATIONS TO ABILITY TESTING

Robert L. McKinley
Mark D. Reckase
University of Missouri-Columbia

Most of the tests presently in use are of the paper-and-pencil variety. Despite this method's common usage, there are a number of problems with this type of testing. One major problem, arising because the same test items are given to all examinees, is that many examinees receive items that are too easy or too hard. If test items are too difficult, an examinee may resort to random guessing or omission of items. If the items are not difficult enough, the test may not be challenging to the examinee. This results in scores containing more error at the extremes of the ability range, indicating that the test is most accurate for average ability examinees. This restricted range of accuracy is reflected in lowered test reliabilities. Other problems that affect the precision of measurement of conventional paper-and-pencil tests include time limit pressures, test security, and differences in test administration (Weiss, 1974).

Computerized testing is the result of attempts to deal with some of the problems of paper-and-pencil tests. One of the more basic types of computerized tests is called computer-assisted test administration (Dunn, Luchene, and O'Neil, 1971). This type of computerized test involves having a computer administer a test, item by item on a computer terminal, in the same order as it would be given in paper-and-pencil format. The examinee responds to the items on the terminal keyboard.

A somewhat more complex form of computerized testing is called computer-assisted test construction (Van Willis and Seely, 1978). This type of test involves the use of a computer to randomly sample, from an item bank, items that have been classified by content area. The computer then prints out the items on a test form, and the test is administered in the traditional manner. One step beyond this is the use of a computer to sample, from an item bank, items classified both by content area and by difficulty. Classification by difficulty usually involves using categories such as 'hard', 'moderate', and 'easy'. All of the above computerized testing techniques primarily help to eliminate problems caused by differences in test administration. However, they still have problems caused by administering the same test to every person--inappropriate items, test security, and time limit pressures.

A type of testing that more effectively deals with these other problems is called tailored testing. In tailored testing, items are selected for administration during the test, rather than before. (For a complete review of the literature on tailored testing see Weiss, 1974.) Tailored testing makes use of computer capabilities to select items during the test that are appropriate for the individual examinee, thereby 'tailoring' the test to each examinee. As a result of this process the precision of measurement is greatly improved. Also, tailored testing eliminates problems with time limit pressures and test security, since no time limits are placed on the test, and each person receives a different set of test items. The purpose of this report, then, is to describe the procedures for computer administration of a tailored test, and to compare the results of a tailored test with the results of a conventional paper-and-pencil test over the same material. The comparison will be made in terms of test reliability, information, and descriptive statistics for both tests. Before beginning, however, it may be helpful to discuss the primary characteristics of tailored testing procedures.

Primary Characteristics of Tailored Testing Procedures

Tailored testing procedures were designed to minimize errors of measurement by administering to each examinee only items of appropriate difficulty. This is accomplished by selecting for administration items that maximize the information about an examinee's estimated ability level. The implementation of this tailoring procedure usually requires computer capabilities. One reason for this is that tailored testing is often based on item characteristic curve (ICC) theory (Lord, 1952; Lord and Novick, 1968), which involves mathematical models of sufficient sophistication as to require the use of a computer. One requirement for the use of the computer is the estimation of certain item characteristics. These item characteristics usually include the item's difficulty parameters, b; discrimination parameter, a; and guessing parameter, c. The calibration of the items in the item pool is usually accomplished by applying one of several existing calibration programs (Wright and Panchapakesan, 1969; Wood, Wingersky, and Lord, 1976; and Urry, 1975) to conventional test item response data in order to obtain estimates of the item parameters.

Another step which requires computer capabilities is the operation of the tailored testing procedures on an interactive basis with the examinee. This is done by a program that selects the items to be administered, administers the selected items, scores the responses, and computes ability estimates. This program is controlled by a number of factors, including: entry point into the item pool, the procedure for estimating ability, the item selection method, and a rule for terminating the test.

There are a number of models available for use in tailored testing. The particular model that has been chosen for this report, the three-parameter logistic (3PL) model, is an ICC model, and falls within a class of models referred to as latent trait models.

The 3PL Model

The 3PL model, as presented by Birnbaum (1968), requires three parameters for each item and one ability parameter for each examinee. The 3PL model is given by

$$P(u_{ij} = 1) = c_i + (1 - c_i)\frac{\exp(Da_i(\theta_j - b_i))}{1 + \exp(Da_i(\theta_j - b_i))}$$

where u_{ij} is the score (0 or 1) on Item i for Examinee j, θ_j is the ability parameter for Examinee j, b_i is the difficulty parameter for Item i, a_i is the discrimination parameter for Item i, c_i is the guessing parameter for Item i, D is a scaling constant equal to 1.7, and $P(u_{ij})$ is the probability that u_{ij} is 0 or 1. (For a full discussion of the assumptions of this model see Lord and Novick, 1968.) The following is a description of the implementation of tailored testing procedures based on this model, and a comparison of the results with the results of a paper-and-pencil test over the same material.

METHOD

Item Pool Construction

The test items that were used for the study were taken from a series of classroom achievement tests that were administered in an undergraduate course on educational measurement and evaluation. Items were taken from six different tests, each fifty items long, covering the content area of educational evaluation techniques. The 3PL item parameters for the test items were obtained using the LOGIST calibration program (Wood, Wingersky, and Lord, 1976).

It would have been desirable to have a large sample of perhaps 1000 examinees to which a single test of 150 items or more could have been administered. However, it is not often possible to administer a test of this length to as many as 1000 examinees at one time. Moreover, for security purposes, it is usually necessary to alter a test between administrations, although there are often numerous items in common from one administration to the next. Because of these limitations, it is generally necessary to combine a series of small sample calibrations. However, separate calibrations of tests will result in item parameter estimates on different scales. This is because item parameter estimates yielded by latent trait calibration programs have invariant scales only to within a linear transformation, due to the arbitrary nature of the zero point and the unit of measurement defined by the calibration (Reckase, 1979). Therefore is is necessary to link the separate calibrations together in such a way as to put the item parameter estimates on the same scale. In this study, linking was done using the Maximum Likelihood Method. Maximum Likelihood linking can be done using the LOGIST program that was used for calibrating the items. In fact, Maximum Likelihood linking using the LOGIST program is essentially the simultaneous calibration of the tests. A full description of this procedure is given in Reckase (1979). This technique, when applied to the six classroom tests used for this study, resulted in an item pool of 183 items. Table 1 summarizes the means, standard deviations, and ranges of the parameter estimates that were obtained from the linked calibrations. The distributions of the item parameter estimates are shown in Figures 1-A, 1-B, and 1-C. For a set of guidelines for tailored testing item pool characteristics see Urry (1977).

Table 1

Descriptive Statistics of Item Parameter
Estimates for the Tailored Testing Item Pool

Statistic	Parameter a_i	b_i	c_i
Mean	.610	-1.674	.180
S. D.	.484	3.361	.010
Low Value	.010	-9.999[a]	.101
High Value	2.001[b]	14.834	.244
No. of Items	.83	.83	.83

[a] This value was an arbitrary lower limit on the 3PL difficulty parameter estimates.

[b] This value is the upper limit imposed by the LOGIST program.

Tailored Testing Procedures

Tailored testing procedures have three main components: a routine for selecting items, a technique for ability estimation, and one or more stopping rules. In this study items were selected to maximize the value of the information function (Birnbaum, 1968) at the current ability estimate. Information was computed using the following equation:

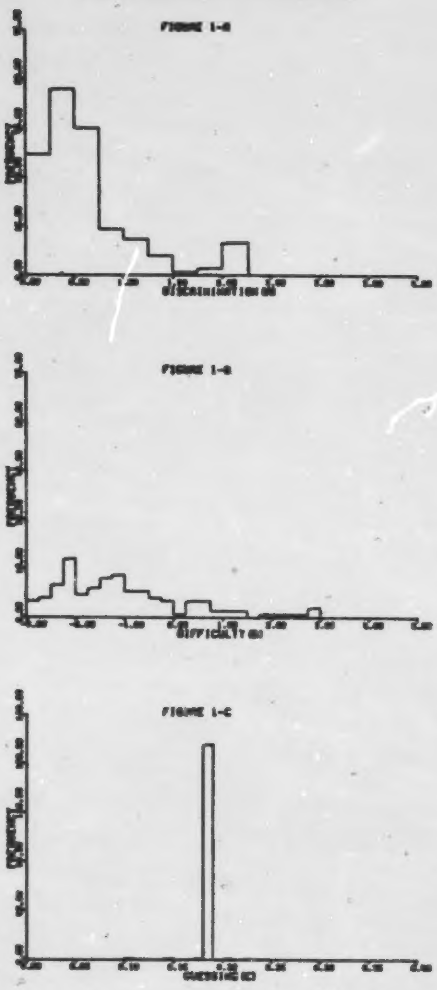
$$I_i(\theta_j) =$$

$$D^2a_i{}^2\Psi[DL_i(\theta_j)] - D^2a_iP_{ij}(\theta_j)\Psi[DL_i(\theta_j) - \log c_i].$$

$I_i(\theta_j)$ is the information for Item i at Ability θ_j for Examinee j, $\Psi(x)$ is the logistic probability density function, $L_i(\theta_j) = a_i(\theta_j - b_i)$, $P_{ij}(\theta_j)$ is the probability of a correct response to Item i given Ability θ_j, and the other parameters are as defined earlier. The total test information, $I(\theta_j)$, is defined as the sum of the item information values:

$$I(\theta_j) = \sum_{i=1}^{n} I_i(\theta_j).$$

ITEM PARAMETER DISTRIBUTIONS



Initial ability estimates for this study were set by random assignment to be either -1.856 or -1.500. These values were chosen so that different initial items would be administered at different testing sessions. They were selected to be close to the median of the item pool so that maximum movement through the item pool would be available in either direction. The first item was selected to maximize information at the initial ability estimate. If that item were correctly answered the ability estimate was increased by a fixed stepsize (.4), and if it were incorrectly answered the ability estimate was decreased by the same fixed stepsize. This fixed stepsize procedure was used until a maximum likelihood ability estimate could be obtained.

Maximum likelihood ability estimation involves finding an ability estimate that maximizes the probability of the response string given the item parameters. This is done by finding the mode of the likelihood distribution. Maximum likelihood estimation requires that the response string contain at least one correct and one incorrect response. A new ability estimate is obtained after each item. Each new item was then selected to maximize the information at the new ability estimate, with the restriction that no item could be used more than once.

Two stopping rules were used for the tailored testing procedure. The tests were terminated when there were no items left in the item pool with information greater than .65, or when 20 items had been administered.

Design

This study employed a test-retest design in order to facilitate reliability comparisons. Examinees took two tests, one week apart, with a different starting point each time. The tailored tests were administered on Applied Digital Data Systems (ADDS) Consul 980 cathode ray tube terminals connected to an Amdahl 470/V7 computer via time sharing option facilities.

Sample

The examinees for the study were volunteers from an undergraduate introductory measurement course. A total of 87 students participated. The tailored tests were administered shortly after a 50 item paper-and-pencil classroom test over the same content. Examinees were told that if either one of their tailored test scores were better than their score on the classroom test, the higher score would be used. They also received extra credit points just for taking part in the study.

Analyses

The research questions in this study included a comparison of test reliabilities and of total test information functions. The reliability for the tailored test was a mixture of test-retest and equivalent forms reliabilities, since the tests in one session were not identical to tests taken in the second session, but some of the items were in common. The total test information comparison was done to obtain the relative efficiency (Birnbaum, 1968) of the tailored testing procedure with respect to the conventional course exam. The relative efficiency of the tailored test was the ratio of the information provided by the tailored test to the information of the traditional paper-and-pencil course exam. The relative efficiency was plotted against ability at selected ability levels across the entire range of ability estimates.

One other analysis performed was a compilation of descriptive statistics for the tailored tests. The descriptive statistics included average test length, average test difficulty, and the distribution of tailored test ability estimates.

Results

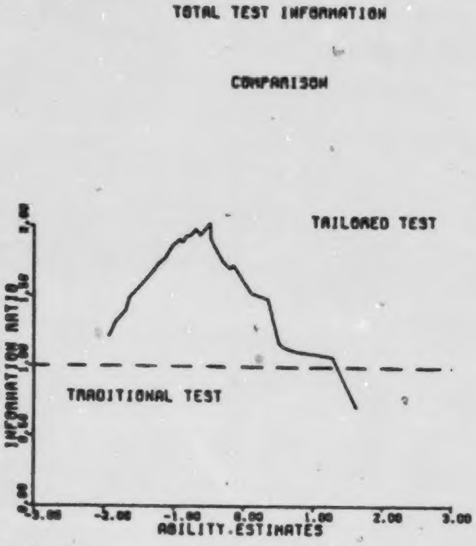
Reliability

The tailored test reliability, obtained by correlating the ability estimates from the first test with those from the second test, was r = .62. The KR-20 reliability of the traditional paper-and-pencil course exam was r = .60. The reliability of the tailored test was actually much higher than the reliability of the conventional test, since normally a KR-20 reliability would be expected to be higher than a test-retest reliability, and the higher reliability of the tailored tests was obtained for a test that was less than half as long as the conventional test.

Information

Figure 2 shows the total test information comparison. The horizontal broken line represents the information of the course exam. The most significant result of this comparison is that, in the ability estimate range of -2.0 to +1.5, the tailored test yielded more information than the paper-and-pencil test. It is important to point out that the tailored test performed best in that range of ability estimates where most of the examinees were classified. See Figures 3-A and 3-B for the distribution of ability estimates for the tailored test.

FIGURE 2



Descriptive Statistics

Table 2 presents descriptive statistics for both sessions of the tailored test.

Table 2

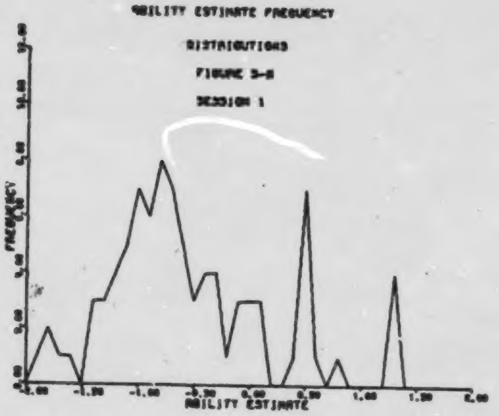
Tailored Test Descriptive Statistics

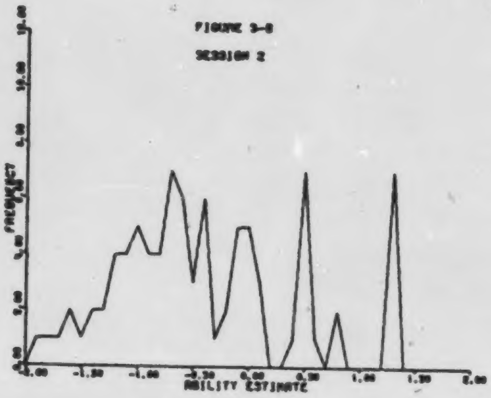
Variable	Test Session 1	Test Session 2
Mean # of items administered	16.23	15.32
Mean # of items correct	12.15	11.71
Mean proportion of items correct	.75	.76
Mean of ability estimates	-.53	-.36
S. D. of ability estimates	.74	.83

A comparison of the statistics for the two sessions shows a great deal of consistency from one session to the next. The mean number of items administered shows the dramatic reduction in the number of test items for the tailored tests, as compared to the 50 item traditional test. The mean proportion of items correct was approximately equal to the ideal proportion suggested by Lord (1952).

Discussion and Conclusions

The purpose of this report was to demonstrate the viability of tailored testing as an alternative to paper-and-pencil testing. It is clear from the results that tailored testing has many advantages. To begin with, the tailoring of tests eliminates the problem of inappropriate items being administered. Moreover, because examinees don't take the same tests, test security problems are eliminated. The average test length is much shorter than a paper-and-pencil test, and time limits are not imposed. Thus, problems created by time pressures are not encountered.





Another advantage of tailored testing involves reliability. In this study a more stringent reliability measure was employed for the tailored tests than for the conventional test. Moreover, the tailored tests were shorter. Despite the shorter length and the stringency of the reliability measure, the reliability obtained for the tailored test was higher than the reliability obtained for the course exam.

Another area in which the tailored tests outperformed the conventional paper-and-pencil exam was total test information. It is important to keep in mind that total test information is the sum of the individual item informations. Therefore, a drop in total test information would be expected when considering a shorter test. Despite this, the tailored tests yielded more information than the paper-and-pencil test in almost the entire ability estimate range. Also, the range of ability estimates where the tailored tests outperformed the conventional test is the range of ability estimates where most of the examinees were classified. Clearly, then, with all these advantages, tailored testing must be considered an attractive alternativ to conventional paper-and-pencil testing.

References

Birnbaum, A. Some latent trait models and their use in inferring an examinee's ability. In F. M. Lord and M. R. Novick, *Statistical theories of mental test scores*. Reading, Massachusetts: Addison-Wesley, 1968.

Dunn, T., Lushene, R. E., and O'Neil, H. F. *The complete automation of the Minnesota Multiphasic Personality Inventory and a study of its response latency* (Technical Memo No. 28). Tallahassee: Florida State University, Computer-Assisted Instructional Center, 1971.

Lord, F. M. A theory of test scores. *Psychometric Monograph*, No. 7, 1952.

Lord, F. M. The relation of the reliability of multiple-choice tests to the distributions of item difficulties, *Psychometric*, 1952, *17*, 181-194.

Lord, F. M. and Novick, M. R. *Statistical theories of mental test scores*. Reading, Massachusetts: Addison-Wesley, 1968.

Reckase, M. D. *Item pool construction for use with latent trait models*. Paper presented at the Annual Meeting of the American Educational Research Association, San Francisco, April, 1979.

Urry, V. W. *Ancillary estimators for the item parameters of mental test models*. Paper presented at the Annual Meeting of the American Psychological Association, Chicago, August, 1975.

Urry, V. W. Tailored testing: a successful application of latent trait theory. *Journal of Educational Measurement*, 1977, *14*, 181-196.

Van Willis, W., and Seely, O. Jr. Computer assisted test construction: two years of user experience with the SOCRATES system. *Proceedings of the 16th Annual Convention of the Association for Educational Data Systems*, Washington, D.C.: Association for Educational Data Systems, 1978.

Weiss, D. J. *Strategies of adaptive ability measurement* (Research Report 74-5). Minneapolis: University of Minnesota, Department of Psychology, Psychometric Methods Program, 1974.

Wood, R. L., Wingersky, M. S., and Lord, F. M. *LOGIST: A computer program for estimating examinee ability and item characteristic curve parameters* (ETS Research Memorandum RM-76-6). Princeton, New Jersey: Educational Testing Service, June, 1976.

Wright, B. D. and Panchapakesan, N. A procedure for sample-free item analysis. *Educational and Psychological Measurement*, 1969, *29*, 23-48.

USING COMPUTERIZED DATA MANAGEMENT TO INCREASE STUDENT RESPONSIBILITY FOR LEARNING IN COLLEGE MATHEMATICS

BY

Dr. Hugh McHenry
Department of Mathematical Sciences
Memphis State University
Memphis, Tennessee 38152

INTRODUCTION

Although a number of forces are pushing for changes, the "lecture method" remains the most widely used instructional strategy for teaching college mathematics. In general the method has three components. 1. Basic concepts which students are expected to learn are presented in lectures by instructors. 2. Following each lecture a set of problems embodying the concepts presented in the lecture is assigned for the student to solve before the next lecture. 3. Periodic tests are given to evaluate the students' understanding of what has been presented. The major role of the instructor is that of lecturer and the role of the student is one of problem-solver. Also, since the lecture is usually the initial and, most of the time, the primary source of information for the student, a large part of the responsibility for learning rests upon the lecturer and how well he does his job.

An instructional research and development project at Memphis State University has been directed at overcoming the following weaknesses in this lecture method for precalculus mathematics. 1. The method does not require an active participation of the student in the initial learning effort on a topic. By listening to the lecture the student is in a passive role and thereby lacks the attention and concentration that promote effective learning. 2. The method does not adequately cope with the situation where the homework is not done. Students will frequently not do their homework. Sometimes it is because of not knowing how to do it; but frequently, it is just because no attempt at all is made. It is difficult and at best awkward to deal with these situations in the lecture system. 3. The method provides very little feedback to the student on his day-to-day performance. Specific reporting to the student is usually limited to the grades on the test papers returned him. These are not on a daily basis and they do not give him a cumulative or summary report on all of his work.

PERFORMANCE LEARNING

The research and development effort has produced a new way of teaching mathematics at the college level. It is called Performance Learning (PL) and emphasizes student participation in all phases of the learning process. In fact, the bulk of the responsibility for learning is put upon the student. PL utilizes the traditional classroom facilities and class meeting schedules without any modification. The computer is an integral part of the new system and is used only by the instructor for information processing.

PL has four major features.

1. The initial learning effort on each lesson is by the student.
2. A written report is submitted by each student at the end of each class period.
3. A comprehensive feedback report is given to the student at the beginning of each class meeting.
4. Support services and facilities are provided for students.

Each of these features is described in more detail in the following paragraphs.

The first feature of PL is to make the student responsible for the first effort at learning new material. It is done by having the homework assignment be the initial learning effort on each lesson. The student's job is to prepare himself to go to class and work out a set of problems.

Audio-tutorial materials are used as the basic course materials. The audio-tutorial textbook is also the course textbook and is purchased by the student in the same way he buys his other textbooks. The Mathematics Department provides the tapes and makes them available to students on a library type checkout plan. Students are encouraged to take the cassette tapes home with them. However, use of the tapes, including checking them

158

out, is optional on the part of the student. One advantage of the recordings is that they retain the lecture component in the teaching program and make it feasible to assign the "lecture" as the homework.

A written report from each student at the end of each class period is the second major feature of PL. The report is created by requiring the student to solve a set of problems which the instructor distributes to the class at the beginning of the class period. All problems are cast in a multiple-choice format so that each student report is a single answer sheet that can be optically scanned for computer input. It has proven helpful to the student to retain the problem sets and collect them in a problems notebook.

To date the problem sets have come from those supplied by the publisher and from a large bank generated by graduate students in the Department. They are duplicated at departmental expense and stored in a file so that instructors have ready access to them. However, this approach is costly and it is very time consuming to develop new sets of problems when textbooks are revised or changed. Efforts are underway, therefore, to use the computer to generate all tests and problem sets on ditto masters. It is also expected that the student will soon be charged a small fee to cover the materials cost of producing the problem sets. The fee would be handled like fees for consumable laboratory supplies in physics and chemistry are handled.

Information flowing from the instructor to the student which tells him how well he is performing is the third feature of PL. Each test and problem set submitted by the student is graded and his answers and grade stored on a computer file. A feedback report in the form of a computer printout is then prepared for each student. It contains a listing of all tests and problem set answers he has submitted along with the grade on each. Right and wrong answers are indicated and a cumulative course grade given. These reports are picked up by the individual students in the classroom at the beginning of each class period and serve two purposes. 1. They point out what was missed and thereby indicate specific problems to be reworked. 2. The comprehensive course grade helps students detect declining performance early enough to do something about it. Also, students near the top of the respective grade categories are motivated to try for a better grade.

The feedback process also generates information for the instructor. Computer listings of individual student grades along with test and problem set analyses are provided. These make the instructor aware of individual students as well as overall class records. For example, if 60% of a class misses a problem the fact is identified for the instructor and he is able to take up the problem with the class at the next class meeting.

Several pilot projects in the early stages of the research on Performance Learning involved individualized instruction and self-pacing. The results consistently indicated that a vast majority of the students could not successfully manage their work in self-paced courses. There was also good evidence to indicate that the improvements observed in individualized programs could be obtained using the traditional lecture method if the instructor spent the same amount of time working with the class as was required to conduct the individualized class.

These conclusions led to the formulation of the fourth feature of PL. It is that a clearly defined program of support services and facilities for students is required. Furthermore, the program should be built around the instructor in such a way as to promote student-teacher interactions. Five things have been found to effectively support PL. 1. Class meetings. 2. Assistance from the instructor. 3. Peer assistance. 4. Audio-tutorial programs. 5. A mathematics learning center. These are discussed in what follows by first describing what goes on in the PL classroom and then relating the support programs to it.

Classes using PL meet with the instructor in the classroom for the same amount of time as regular traditional classes. Each class meeting begins with the instructor placing the problems and answer sheets on his desk at the front of the room. The reports for the students are divided into three groups and placed on three desks across the front of the room. Each student then comes by and picks up his report and a copy of the problems and returns to his seat. This takes two or three minutes in a class of 40 people.

When picking up the reports is completed the instructor discusses with the entire class any matters that he judges to merit consideration. Such things might be a problem missed by most of the class, or a mini-lecture on a topic that is causing difficulites, or giving the results of a major exam. Not more than one-third of the class meetings include group sessions of this type, and then not more than ten minutes are usually used for them.

The student begins work on the problems for the day when the opening group session is over. If there is no opening business, he goes directly to work on the problems he has just picked up. While doing them he is encouraged to seek help from the instructor, who stays in the classroon, or from fellow students. He also uses his textbook. This means that the daily problems are not treated as texts but as learning experiences. The answers are recorded on the answer sheet and turned in to the instructor when the student is finished. He is free to leave class when his report is in.

One purpose of the class meetings is to help the student maintain a regular work schedule. The problems in class each day motivate him to do the lecture homework assignments as well as provide practice and review on what is covered in the homework lecture. Turning in the daily reports to the instructor has proven to be an effective way to get students to feel obligated to complete each lesson. Therefore, the class meetings are an integral part of a PL program.

Peer assistance in doing the problems has developed into a major feature of the class meetings. Students ask questions of one another much more readily than of the teacher. The result is that they get the help they need more frequently than when the only source is the instructor. Also, if several of them in a group do not know the answer to a question, they call the instructor to their group for help. This group setting is not threatening or embarassing to an individual. The peer groups are formed by the students as they establish working relationships that are beneficial to them.

Since the instructor is in the classroom at all times for the purpose of assisting students, he spends almost all of his time talking to individuals and small peer groups. A close working relationship is readily established with a large portion of the class and students frequently express appreciation for it.

In addition to the use of audio-tutorial materials that has already been described, a mathematics learning center operated by the department provides support for PL. It handles the tapes; but more importantly, it helps students with their mathematics. The center provides a free, walk-in service in answering questions and helping solve problems for any student enrolled in mathematics at the University. It is open from 8:00 a.m. to 7:45 p.m. Monday through Friday and is staffed by graduate students. Therefore, assistance is easily available to students when they are away from the instructor and the classroom.

Testing and grading are emphasized in PL. The testing is based upon the premise that frequent testing promotes learning. Therefore, a major examination taking an entire class period is given every two weeks with a two-hour comprehensive final coming at the end of the course. The major tests make up 65% of the total grade, the final 15% and the class problems 20%. A grading scale is set in advance of the beginning class meeting and given to each student in writing. The same scale is used for all of the tests, the daily problems sets, the final, and the course average. This way the student has a clear indication of where he stands at all times.

It is assumed in PL that a computer will be used by the instructor for grading, storage of records, and preparation of reports. However, it is not required that a teacher have prior experience with a computer. An orientation session prepares him to use a system of hardware and programs to process large amounts of test-type data quickly and easily.

All of the instructor's work is done at a time-sharing terminal. The one currently in use consists of a Texas Instruments Silent 700 thermal printer interfaced with a Datronics 5500 Test Scoring Machine. The Test Scoring Machine handles the optical input to the computer. A Xerox Sigma 9 main-frame computer is being used. Large volume printing tasks, such as the student reports, are set up by the instructor at his terminal but are executed in the batch mode and printed on a line printer at the rate of 300 to 900 lines per minute. Small printing jobs, such as tests analyses, listing of class rosters, or one student's record are executed at the instructor's terminal. This terminal is housed in a converted classroom adjacent to the learning center. It is used only by instructors and only for information processing in Performance Learning.

The software package is one called Computer Assisted Testing Services (CATS) which was described in a presentation at the 1978 AEDS meeting in Atlanta (McHenry, 1978). CATS is a set of BASIC computer programs designed to facilitate the use of multiple-choice tests in eduation. The package is an interactive, time-sharing system operated by the instructor. CATS 1. Scores Tests, 2. Stores tests scores and answers, 3. Has numerous options for editing and managing records, 4. Provides for retrieval of data for students, instructors, and researchers, and 5. Generates test analyses. A special language containing 15 catagories of commands using conventional English words and systax is used to operate the system. Each command is issued in response to the system prompt 'WHAT NOW?'. More specifically, CATS consists of a master control program and a set of service programs which act as subroutines. Altogether there are 23 separately stored programs and 5 permanent data files, all of which are maintained on public disk storage.

160

RESULTS

The first significant break in the research and development work leading to the formulation of Performance Learning came in the Spring Semester of 1978. It was in a college algebra class taught by the author and was the first time the lecture was made the homework. Of the 27 students who enrolled in the class 85.2% (23) completed the class and 81.5% (22) made grades A through D. Corresponding statistics for the Department in precalculus courses are 60% for completion and 50% for passing.

A total of 485 students in 14 classes from three courses subsequently used PL between January, 1978 and December, 1979. Ten classes were in college algebra, three in trigonometry, and one in introductory algebra. Five different instructors have been involved. Three were full-time tenured faculty members with an average of about 15 years college teaching experience and two were graduate teaching assistants. One assistant had taught the semester prior to using PL while the other was a foreign student with no prior teaching experience and a language problem.

Course completion and grade distribution data were gathered on each of the 14 classes. In addition, a controlled experiment was conducted in college algebra during the Spring Semester of 1979 in which data amenable to statistical analyses were collected. Three instructors each taught a class using the traditional lecture method and a class using the PL method. Pre and post measures were made on attitude and achievement. Records were also kept on course completions and grade distributions for the traditional classes as well as for the PL classes. The instructors in the experiment were the two graduate teaching assistants identified in the preceding paragraph along with the author of this report. The classes were formed by the usual registration process. However, no instructors were identified in the schedule of classes for any college algebra classes in order to prevent students choosing a class because of the instructor. Teaching assignments were handed out the night before classes were to begin the next morning.

The observations and data gathered from these uses of PL can be summarized under the following five headings.

1. Course Completions
2. Test Performance and Grades
3. Attitudes
4. Tape Usage
5. Subjective Observations

One way to judge Performance Learning is by how well it motivates students to persist in their classes. It is common practice for some students to formally drop from class while others just stop attending. Therefore, a student who stays in a class to the point of taking the final test can be said to generate one course completion. TABLE A reports the course completion rate for the 485 PL students along with the rate for the Mathematics Department in the Fall Semester of 1979. The Departmental figures are based upon class records from 8 instructors in 14 classes in precalculus mathematics. Two of the teachers are past winners of the Outstanding Teaching Award for University.

TABLE A. Percent of Students Completing Their Classes

Class Completed	Traditional Classes	PL Classes
	N = 546	N = 485
Yes	61.0%	79.8%
No	39.0%	20.2%

$\chi^2 = 43.12$ $\chi^2_{.05} = 3.84$

A Chi-Squared test for independence of course completion and class type yields the values given at the bottom of the table. Since $43.12 > 3.84$, the hypothesis of independence is rejected in favor of course completion being dependent upon class type. The nature of the dependence is indicated by the fact that the 79.8% rate for PL is almost 19 points better than the 61.0% for traditional classes. That is, the course completion rate for Performance Learning is significantly better than that for the traditional lecture method.

Course grades and performance on tests are among the critical variables upon which PL must be evaluated. Only data from the controlled experiment are used in this evaluation in order to control the teacher variable and have comparable data on the two ways of teaching. Since each of the three teachers taught both a traditional and an experimental class, the instructor variable is held fixed in comparing the teaching methods. Common tests and grading standards were used in both types of classes. This created comparable data upon which to make comparisons.

TABLE B gives the percent of students making passing grades for both ways of teaching. A Chi-Squared analysis yields $6.00 > 3.84$ and indicates a significantly higher proportion of A through D grades for Performance Learning than for the control classes.

TABLE B. A Comparison of Course Grades for Performance Learning Classes and Traditional Classes.

Grade	Traditional Classes	PL Classes
	N = 124	N = 113
A - D	45.2%	61.1%
other	54.8%	38.9%

$\chi^2 = 6.00$ $\chi^2_{.05} = 3.84$

A 40 item multiple-choice test designed to be used as a final examination in college algebra was given to each student involved in the experiment. The test was administered as a pretest during the second class meeting and then as one-half of the final test. The gain from the pretest to the post-test is used as one criterion variable in comparing the teaching strategies.

Also, a "traditional" type test on which students responded to a series of questions by solving problems and writing up their answers was given to each student as the other half of the final test. These were graded by the instructor. The gain from the pretest to this post-test is also used in the comparisons.

TABLE C reports the means and standard deviations for these various tests and gain scores. The results are given for each of the three instructors as well as for the pooled group. The class sizes ranged from 12 to 35. The pooled groups for Performance Learning had 66 while the control group had 60. All of the tests were graded on the basis of 100 points. Therefore, the values in the table are percentage scores.

TABLE C. Test Results from the Performance Learning Experiment in the Spring Semester of 1979

Teacher	Class		TESTS						GAIN SCORES			
	Type	Size n	I*		II*		III*		I@		II@	
			$\bar{X}$	S	$\bar{X}$	S	$\bar{X}$	S	$\bar{X}$	S	$\bar{X}$	S
1	CTL[1]	18	32.4	12.1	47.1	11.8	48.7	25.1	14.7	16.3	16.3	25.7
	EXP[2]	19	28.3	13.2	54.6	19.2	36.9	29.3	26.3	21.7	8.6	31.2
2	CTL	24	28.5	13.2	52.1	14.0	63.4	20.8	23.5	12.5	34.8	19.6
	EXP	35	25.8	9.0	52.5	18.2	51.4	22.2	26.7	15.6	25.6	20.1
3	CTL	18	27.2	11.8	50.0	14.2	57.6	14.5	22.8	8.5	30.4	9.6
	EXP	12	21.3	11.9	46.7	17.4	42.3	19.5	25.4	17.8	21.1	21.7
Pooled	CTL	60	29.3	12.4	50.0	13.4	57.2	21.2	20.7	13.2	27.9	20.7
	EXP	66	25.7	11.0	52.1	18.4	45.6	24.6	26.4	17.7	19.9	24.8

* Test Identification
I: 40 Item Multiple-Choice pretest, 50 minutes time limit
II: Same as I used as one half of final two hour test
III: Traditional type math exam where the student solves the problem on the test paper showing what he did

@ Gain Score Identification
I: Test II - Test I
II: Test III - Test I

1 CTL: Control class taught by the Lecture Method
2 EXP: Experimental class taught by the PL Method

Gain Scores I were obtained for each student by subtracting the pretest score from the post-test score on the multiple-choice test. This was done to correct for any entering differences between students in knowledge of college algebra. Comparisons of the experimental and control groups for equality of means and homogeneity of variance were made using statistical methods for the pooled group.

An F-test for homogeneity of variance produced $F = (17.7/13.2)^2 = 1.79$ with 59 and 65 degrees of freedom. The critical value for this test is $F_{.05} = 1.67$. Thus $1.79 > 1.67$ suggests that the gains in PL are more variable than those in the traditional classes.

The appropriate t-test for equality of means when the variances are not equal gave $t = 2.06$ with 121 degrees of freedom. The critical value is $t_{.05} = 1.98$. Thus the 5.7 points by which the PL mean gain exceeds the control group mean is a statistically significant difference since $2.06 > 1.98$. This type of difference is further supported by the fact that the PL mean gains were higher than those for the control group for each of the three instructors.

Of the three instructors who participated in the experiment, the first one listed in TABLE C was a graduate teaching assistant in her second semester of teaching. She was openly opposed to the PL concept before being assigned to the project. The second instructor was the author who has more than 20 years of college teaching experience. The third was a foreign graduate teaching assistant with an English Language problem. This was her first teaching assignment.

An analysis of variance comparison of the control group means for the three teachers yielded $F = 2.76$. The critical value is $F_{.05} = 3.15$. Thus, $2.76 < 3.15$ supports the conclusion of no differences in teachers. A visual comparison of the PL means for Gain Scores I indicates that they are extremely close to one another. Hence, no significant differences in teachers were found in Gain Scores I.

Since all tests and problem sets in PL are in multiple-choice format, students are not required to write-up solutions to problems. In order to study the possibility that PL does not adequately teach students to write in mathematics, one part of the final exam was a traditional test in mathematics where each student had to solve a series of problems and write-up his answers to be graded.

The Gain Scores II in TABLE C are the result of subtracting the pretest score for each student from his traditional test score. Again, gain scores were used in order to compensate for any entering differences in students. Statistical tests for equality of means and homogeneity of variance between the two treatment groups were made for the pooled groups.

For equal variability, $F = (24.84/20.74)^2 = 1.43$ with 65 and 59 degrees of freedom was found. This does not indicate a significant difference in variances ($F_{.05} = 1.53$).

A t-test for equality of means gave $t = 1.96$ with 124 degrees of freedom and $t_{.05} = 1.98$). Although this does not indicate a significant difference between the experimental and control groups, the control group was ahead by 8 points. Also, the control group was ahead of the PL group by 8 or 9 points for each of the teachers. The size and consistency of this difference would seem to merit further investigation of the possibility that PL does not adequately teach students to write in mathematics.

Student attitudes were also investigated by administering the Mathematics Attitude Scale (Aiken, 1963) during the first class meeting and again during the last class period. A criterion variable was created by subtracting the pretest score from the post-test score. TABLE D summarizes the means and standard deviations for each teacher as well as for the group formed by pooling the students from all three teachers.

An almost universal pattern of a decline in mean attitude values is indicated by the table. The only exception is for the PL group under the second instructor. A statistical comparison for the pooled group shows homogeneity of variance with $F = (17.79/15.21)^2 = 1.37$ and $F_{.05} = 1.67$. However, a t-test for equal mean values yielded $t = 2.70$ where $t_{.05} = 1.98$. This supports the conclusion that a significant difference exists between the two ways of teaching. The PL group declined 2.1 points as compared with a 4.7 points decline for the traditional classes. Thus Performance Learning appears to create better student attitudes than the Lecture Method.

TABLE E gives data on the use of the recorded lectures available on cassette tapes. The information was collected from three different classes during the Fall Semester of 1979. One class was in introductory algebra. Another was in college algebra and one in trigonometry. All had a combined total of 112 students.

While the large number of small frequencies in the table prevents detailed statistical analyses, the data show that 42.2% of the students never used the tapes.

TABLE D. Attitudes of Students Completing Their Course in the Spring of 1979

Teacher	Class Type	N	Mean			Standard Deviation		
			Pre	Post	Difference	Pre	Post	Difference
1	CTL[1]	18	61.1	53.1	-8.0	16.9	16.3	17.5
	Exp[2]	19	56.3	51.4	-4.9	18.7	21.3	14.7
2	CTL	24	59.0	54.6	-4.4	18.0	17.4	19.1
	Exp	35	36.0	60.4	.4	15.9	19.2	15.2
3	CTL	18	62.4	60.4	-2.0	16.1	20.3	16.8
	Exp	12	53.3	46.6	-6.7	17.5	17.9	15.1
Pooled	CTL	60	60.6	55.9	-4.7	16.9	18.0	17.8
	Exp	66	57.5	55.3	-2.2	16.9	20.1	15.2

1. CTL: Class taught using the Traditional Lecture Method.
2. Exp: Class taught using the Performance Learning Method.

That is, 51.8% were users. Furthermore, 33.0% were light users while 18.8% were heavy users. These figures suggest that the tapes are indeed an important part of the PL method.

TABLE E. Cassette Tape Usage and Course Grades

Course Grade	User Type		
	Non[1]	Light[2]	Heavy[3]
A	1	2	4
B	13	8	7
C	21	13	8
D	4	1	1
F	13	7	1
Quit Class	2	6	0
Total	54	37	21

1 - Non User: A student who never checked out one of the tapes.

2 - Light User: A student who used the tapes on less than half of the lessons.

3 - Heavy User: a student who used the tapes on at least 50% of the lessons.

In addition to these quantitative results that have been given, three conclusions are suggested by the teachers' experiences. 1. Reading the mathematics textbook is not a serious problem in this way of teaching. This is based upon the fact that 48.2% of the students never used the tapes; yet, 72.2% of these non-users passed the courses. Also, the overall success patterns deny serious difficulties in reading the textbook. 2. Students form peer working groups on their own during the first three weeks of the course. Most of these will essentially remain intact for the remainder of the course. 3. Instructors without prior computer experience can use the system without much difficulty. It was done by 2 out of the 5 teachers mentioned in this report. One of them was instructor 3 referred to in the tables.

In summary, Performance Learning is a way of teaching precalculus college mathematics which requires the student to assume most of the responsibility for learning. Evidence indicates that it significantly improves course completion rates and achievement. It also has less of a negative impact on attitudes than does the traditional lecture method. It may not teach writing skills as well, though.

REFERENCES

Aiken, L.R., Personality Correlates of Attitude Toward Mathematics. The Journal of Educational Research, 1963, 56, 476-480.

McHenry, Hugh, Computer Assisted Testing Services. Proceedings of the 16th Annual Convention of the Association for Educational Data Systems, 1978, 274-278.

DEVELOPMENT OF A DISTRIBUTIVE EDUCATION SYSTEM

Lou Molettiere[1], Benn Konsynski[2], and
Jack Stott[2]
[1]American Telephone and Telegraph
Piscataway, New Jersey
[2]Department of Management Information
Systems, University of Arizona

ABSTRACT

Important factors are presented in the design of a large scale distributed information system for "distributive education." The long range effort involves the use of powerful microprocessors controlling videodisc and other communication devices. A migration plan includes the centralization and integration of currently diffuse educational delivery and support systems. The basic administrative and educational functions are described.

INTRODUCTION

We are witnessing an extremely rapid rate of change in technology in our society. This has imposed new and expanding demands for education. It has also created a great demand for more effective training programs in industry, where time off the job is critical. This paper examines some of the means of provision of more extensive and effective training to improve the productivity of industrial training organizations in the 1980s.

CAI/CMI PRINCIPLES

With an increasing need for better and more educational programs, instruction developers have been studying and developing new technologies aimed at two goals. The first goal is increasing the amount of knowledge that the student learns during participation in an instructional program, and the second is increasing the number of students that can flow through an educational program.

In the last decade, many industrial organizations have started using the computer to achieve these goals. The common terminology for this is Computer-Aided Instruction (CAI), and Computer-Managed Instruction (CMI). Many CAI/CMI systems are used in delivery of training within the Bell System. The resulting consistency and economic problems are being addressed through long range planning of an integrated system.

Three major themes have emerged. The first concerns the individualization of instruction. It has been noted that students learn in different ways and progress at different speeds, so that an instructional program that uses a single approach and proceeds at a single rate of progress will be of limited effectiveness.

Individualization is not only beneficial to the student, but can also benefit the educational organization: students can work at other times than those that coincide with the schedule of the instructor, or the schedule of an entire class or other educational resources. This is an important factor in dealing with industry training that might pull individuals off the job.

The second theme is one of behavioral objectives. This concept involves taking a curriculum, or subject, and performing a task analysis on it to develop a set of "behavioral objectives." These behavioral objectives are the expected outcome of the course of instruction expressed in terms of the desired behavior or performance of the student. These techniques are important in the development of training development standards.

The third theme is one of educational technology. The intent is to expand the nature of educational technology past the point where CAI is merely automated programmed instruction. The goal is to use the special capabilities of the computer to teach in the most cost effective and productive manner. The Bell System is currently making use of the many CAI systems currently available (e.g. PLATO) and is presently integrating control of these systems to facilitate an orderly evolution.

COMPUTER-AIDED INSTRUCTION (CAI)

CAI can be defined as any use of the computer in order to convey knowledge to the student. There are many modes of CAI and within these modes as many methodologies as there are implementors.

The first mode is problem solving. In this mode, the student uses the computer to solve problems that would take many hours or excessive computation to solve by hand. This mode opens a new realm of problems and learning to the student. Statistical models, optimization techniques, and other applications to solve a particular model are examples of this mode.

The second mode is drill and practice. The computer is very well suited for this repetitive task. Where instructors would not use their time wisely for conducting a student drill, the computer is an obvious alternative.

The third mode is inquiry. In this mode, a data base of information is developed which the student can query for facts about a topic. An example of the inquiry mode is PIP, Product Information Package, a data base of product information that marketing personnel can use to keep up to date on tarrif status and product options.

The fourth mode is simulation or gaming. In this mode the students participate in a simulation of a real world event, such as the management of a business. Students in teams make business decisions and the computer simulates their effect. AT&T Long Lines has recently made use of this technique in the training concerning business functions and business decision trade offs.

The fifth mode is tutorial instruction. In this mode, an instructor writes a course that presents information and examples, and then quizes the student on the information. Many of the commercial packages used by Bell facilitate this type of interaction (e.g. primitive PLATO courses).

The sixth mode is author mode In this mode the machine uses algorithms and formulas to generate questions to present to the student. The course author supplies subject primitives, algorithms and decision criteria and the computer can then generate random exercises for the student to use.

The curriculum contents in the industrial environment are very skill and knowledge oriented and usually directly related to job function. A task analysis usually serves as the basis for the skill and knowledge determination. Criterion are established for evaluation of satisfaction of requirements. The detailed procedures in design and development of effective training have also been applied to the CAI delivered subcomponents.

COMPUTER-MANAGED INSTRUCTION (CMI)

Computer-Managed Instruction, CMI, can be divided, for purposes of this discussion, into two major parts: internal and external to the instruction.

Internal CMI deals primarily with the testing cycle of test construction, delivery, response collection, response judging, and feedback of results. Also included are programs that analyze the results and prescribe a course of action to remedy any deficiencies. These CMI functions are tied closely to the CAI aspects, and are, in fact, an integral part of the course. On the other end of the spectrum, there are several training organizations that use test construction and grading systems that are totally independent of the method of instruction. These usually serve primarily as time saving devices for the instructor.

The strategy taken with the CMI has been correlated with the instructional philosophy used. Some alternative strategies that have been considered are forced mastery, required or optional testing, retesting, testing only the un-mastered topics, allowing a maximum number of attempts, continued answering until correct, and early test termination based on some criteria. Our studies have shown that it is difficult to separate them from the CAI strategy.

External CMI functions are comprised of systems to schedule resources such as classrooms, instructors, materials, or terminals, systems to generate reports for instructors or administrators on tests, students, or resources usage, and systems for general accounting for resources. There are several modules useful to the manager of an industrial training center. These modules are primarily external CMI functions, but could easily be integrated with CAI or internal CMI programs. The modules include:

Course development monitoring
Personnel management
Pre-course assistance
Feedback and evaluation
Student tracking through the curriculum
Course comparison

A system supporting these functions in an integrated form is being implemented.

TRENDS AND NEW DIRECTIONS

The current CAI/CMI emphasis is toward development of systems that are more functional than programmed text on the machine. Strictly programmed systems may be more of a hindrance than a help to the student. The new systems make the primary sources of knowledge directly available to the student through organized files of information and procedures. The student is given the necessary skills of information management, computation and composition. New and more advanced kinds of equipment are used more widely and the responsibility for greater skill in information acquisition and decision making fall on the student.

For some students and certain curricula, the effectiveness of a computerized, author-controlled presentation are greater than the use of text-books, audio-visual aids, and typical classroom instruction, and justify the increased cost per unit. The functions of informing personnel on new and proposed products are handled more effectively and correctly through a centralized, computer-based distribution system. The sales personnel are informed on products, options and tarrif status.

Special purpose programs have been written to assemble instruction materials from elements of a subject and relationships among these elements. For example, the sales personnel can request information on a product and its relationship to other products.

Communications between networks play an important role in permitting the preparation and use of computer-based exercises within existing economic contraints. Linking computer networks allow a greater number of students to access the CAI material, which decreases the cost per student of development. The system facilitates the communication between marketing, staff and other departments in the Bell System.

Bell is considering the development of a distributed education network involving Regional Education Center interface, through multiple networks. This facilitates Regional migrations. Other advantages are quick dissemination of material, distribution of use to a growing audience, more lessons developed with more people in contact with the system, large scale comparative testing capabilities, and increased quality of training.

The student is studied by the system which keeps statistics on the responses, and evaluates the best plan of action for each particular student. This is accomplished in conjunction with control education, Regional Training, or Local (Field Sales) Training organizations.

A major trend in design of computer based exercises is a shift from program to learner control. The designer is putting more effort into providing information from which the student derives his own diagnosis, and into making available alternative interpretations and guidance from which the student assembles his specific prescriptions. The student works at his/her own pace and time, facilitating less "road time" away from the office and promoting a clearer comprehension. This training does not preclude the centralized training necessary for concepts, advanced products, sales, etc.

Several changes in technology are changing the current methods of CAI. Low-cost and easy to use micro-computers are providing access to highly interactive computing including graphics. Video techniques, such as the micro-computer based video-disc player, are enhancing graphics and low-cost storage for instructional computing. With the video-disc, a system may be developed that has computer programs, stored data and audio-video images all on low cost storage medium. This will allow a tremendous increase in CAI program flexibility and portability.

Very low cost and very large memories are being used in educational time-sharing systems. These technologies drop the cost of systems significantly. As Bell is a leader in development of many of these technologies, the Bell System will be among the first to benefit from the economies that will be encountered.

More futuristic advancements will be facilitated through development of sound and thouch sensing interfaces. Computer systems that can understand voice input will allow very user friendly programming.

The mechanized advancements will not prove nearly as valuable in the shorter run as the new teaching techniques that CAI allows. These techniques enhance individualization to a level not widespread today. These programs need large memories and, in many cases, dedicated machines, and require very large development efforts. The result, however, is an increase in the instructional effectiveness to levels never before thought obtainable.

CONCLUSION

As discussed earlier, many CAI/CMI systems have been studied in an effort to facilitate long range planning for an evolutionary, distributive education system for the Bell System. A wide range of systems have been studied by or used by the Bell system. These include: TIPS, CMS, DECAL, PLATO, AUTHOR, ACES, EPIC, Coursewriter, TICCIT, IIS, PLANIT, etc. A long range plan calls for the utilization of the advanced technologies that are almost here. The long range integrated system will take advantage of microprocessor and videodisc technologies as well as the advanced communications technology available to the Bell System.

Distributed Data Processing
at the
Wayne County Intermediate School District

Larry Molloy
Support Consultant
WCISD
33500 Van Born Road
Wayne, Mich 48184

INTRODUCTION

The shift from a centralized to a distributed system is, at first glance, often less than impressive to the end user. Out goes a card reader and a printer. In comes a tube, a printer, and a box in the middle. The jobs that were once punched on cards are now entered on tubes, with their fair share of confusion. The success rate climbs rapidly though, and the tube begins to do more than act like a card reader. With some new training and new manuals the box in the middle begins to offer some of the things promised. Finally, the end user and the box become close friends. The box is recognized for what it is; a remote processor linked to the central computer. The end user becomes more operator than clerk, one in control of his own processing destiny. The preceeding scenerio is not fantasy. It has happened and is happening in the school districts served by the Wayne County Intermediate School District. As an agency defined by the State Board of Education and separate from normal county government, the Intermediate serves 36 local school districts. The constituent districts conduct classes at the elementary, secondary, and community college levels for more than 450,000 students as the third largest school district in the nation. The Intermediate's mission is to promote student learning by providing services to local school districts which will improve instructional programs and school system operations. Such services include in-service teacher training, curriculum evaluation and modernization, occupational and career education consultation services, research, and data processing services. The data processing services include student related services (grade reporting, student attendance, scheduling, etc.,) management services (payroll, finance, etc.) and computer based education (student programming, simulation, computerized guidance, etc.).

BDP: BEFORE DISTRIBUTED PROCESSING

For many years the Intermediate offered the above services in the same manner as other data processing organizations. An IBM 370/158 AP was used to serve the data processing needs of 27 local school districts. Each district had one or more remote card readers and optical scanners for input and at least one remote printer for output. The six megabytes of main memory and 8 1/4 billion bytes of on-line disk storage allowed each user to submit their jobs according to their own prioritiies. The data processing staff was organized into the following groups:

1. Operations; whose task was to run the computer and start communication lines when necessary and handle user calls regarding their terminals.
2. Systems; who kept the machine operating properly and had little interface with the end user.
3. Student Services; consultants who specialized in the use of student related software and handled user calls.
4. Management Services; consultants who specialized in the use of management and financial software and handled user calls.
5. Special Projects and Maintenance; whose task was to repair ASCII devices, develop hardware for special needs and write special computer programs to be used by the Intermediate School District.

The hardware and software functioned well. The users were receiving the reports they desired. If a user had a problem it was easy to determine whom to call. Operations took care of the computer, consultants handled special projects and maintenance fixed terminals and scanners. But success breeds growth and growth can often bring problems. The demand for additional remote units grew at a rate that could have easily overwhelmed the existing computer. The additional units would also have created a jungle of phone lines, all leading to one location.

WHY DISTRIBUTED PROCESSING?

If the Intermediate were to continue to serve its constituents in an efficient and economical manner a plan had to be developed to accomodate growth. After examining several alternatives a decision was made to shift from a centralized operation to a distributed one involving a host computer and local processors. Several factors favored this plan. Perhaps the most important was an analysis of the goals of the end user. While some jobs, such as student scheduling, require the capability of a large system, others, such as updating and displaying revised financial accounts could be done locally if the updated data were stored for future access. It was evident that processing some data at local sites was not only feasible but desirable. It would provide room for growth without replacing the existing central system and would also allow the development of new applications that would increase the effective use of local data. Distributed processing also offered flexibility. The local processor could support terminals directly or provide them with a link to the central system. Students, counselors, teachers, and administrators would be able to share the local equipment. Local and host data base design allows the user to edit data locally before transmission to the host. It also allows the user to store and display data locally at any time, even when the host system is down for maintenance.

THE MOVE BEGINS

To test the usefulness of distributed data processing in an educational environmnent a pilot project began in December 1977. An IBM 3790 was installed in the 4,600 student Woodhaven School District. The ease of use of the system, the local editing and control functions, and the use of multiple video tubes attached to the local processor contributed to the success of the pilot. By 1979 five 3790's had been installed in local districts. When the new IBM 8100 was announced a decision was made to replace all 3790's and to provide the opportunity for each local district to acquire an 8100. The first 8100 was installed in August 1979. The process to install units in all of the remaining districts will occur over a two year period. The individual districts will determine their conversion schedule based upon their resources and staff training. The installations are proceeding smoothly and each inevitably produces a new advocate for distributed data processing. But what did it take to support the end user during the shift to distributed processing? The impact of the shift may have been as large on the Intermediate's data processing organization as on the end user.

IT STARTS WITH HARDWARE AND SOFTWARE

Assuming the money is available, it isn't that hard to acquire hardware or software. The vendor eagerly supplies the proper boxes and tapes. But since the Intermediate was one of the nation's first educational 3790 and 8100 users, the vendor graciously provided an additional component: support staff. This was extremely beneficial because new systems too often behave like the Christmas toy with the 'insert tab A into slot B' instruction set. It really should work but it doesn't quite do what it's supposed to do. The Intermediate's staff spent time huddled with vendor staff to identify and solve problems. In a centralized operation the problem was either at the central site, perhaps a bad phone line, or maybe a bad reader-printer. But now the chance for error grew. Was it a local or host software or hardware error? Was the communication line bad between the host and the local processor or between the local processor and local tubes? Were jobs sent to the host ever received? As local districts implemented distributed processing new situations arose that created new problems.

HELP

Eventually the puzzles are solved and the system works. But a curious change had taken place in the users. Their close communications with the systems group involving the local processors formed a bond that extended to all problems. Soon the systems group was taking calls about problems with applications. They would verify that it was not a system or distributed processing problem and then work with the appropriate consultant to solve the problem. This new role was unique and it had not been anticipated. The systems group was looked on by the distributed processing users as the source of help for all problems. This not only over-burdened the systems group but also prevented the support consultants from doing their job effectively. A way had to be found that would handle the users' problems in an efficient manner and restore order to the roles of the various departmental groups.

A series of staff meetings were held with outside consultants to define the goals of the department and tne roles of the various units. These meetings led to the reorganization of the department and the birth of a new feature.

THE HELP DESK

The biggest and most apparent outcome of the reorganization was the "Help Desk". Rather than being one physical desk it actually became one phone number that rings to either of two desks. The desks are in a room known as the network control center. It is staffed by two staff members. One is primarily responsible for network and communications problems while the other has an applications background. Their task is to receive all user calls and log them by date, time, caller, problem description, problem number and assignments. If the problem can be solved during a short initial phone call it is noted on the log. If not it must be referred to the appropriate staff member. Since the distributed network involves a variety of problems at many levels of difficulty, all staff members, not just consultants must be used as available resources. Primary and secondary responsibilities were assigned in the areas of applications, systems, communications, operations, and maintenance. Cross training was encouraged to insure that there would always be someone available to assist a user. The Help Desk is not designed to solve problems but to diagnose the problem and refer it to the proper person. This is done by listening to the user, noting the problem, promising to have their call returned within 10 minutes, and then submitting a written description of the problem to the appropriate staff member.

One problem in returning the call within 10 minutes is to locate someone who has primary or secondary responsibility in the area. To assist this effort a new sign out procedure was established in the network control room for all staff members. While this was a bit more red tape than in the past, it was evident that the Help Desk could not function without knowing where staff members are at all times. Once notified of the problem that particular staff member is to call the user. The problem may or may not be settled with that call. Time and tests may be necessary and are expected. Upon solution of the problem the staff member must return the problem description and its solution to the Help Desk. This assists in creating a log of open problem as well as educating Help Desk staff in the solution to common problems. Reports are also generated on the type of problem, by application, by system, by equipment, and by district.

GETTING STARTED

A Help Desk will work if people call it. It is difficult to break old patterns so the staff had to help the users. All staff members were told not to repond to any user problem calls directly without a Help Desk assigned problem number. Calls were re-routed from the consultant, to the Help Desk, and back to the consultant. This was awkward initially but it did create a communication pattern that has aided both the staff and the user in problem solution. However, habits are hard to change and not all users adopt new habits readily. A concerted effort was made by all staff members to route any stray calls back to the Help Desk for proper sequencing. The individuals staffing the Help Desk were at the trainee level. They were not expected to know the answers to all questions, especially when the project first began. Their first weeks were spent receiving calls, writing problem descriptions, and promising to have someone return the call. But after a while they began to notice patterns of problems. The staff's notation of how a particular problem was solved gave the Help Desk staff an awareness of how similar problems could be solved. In other cases, such as maintenance problems, a check list was developed to assist in problem solution. For example, if a terminal was not working, was it plugged in, was it turned on, etc? The variety of problem calls and the notes on their solution provided the Help Desk staff with a broad understanding of the Intermediate's data processing functions. To provide other staff with an understanding of the Help Desk, support consultants and others took turns at staffing the desk one day a week. This not only provided understanding but also valuable cross training.

AN UNEXPECTED BENEFIT

The Help Desk has been successful in reestablishing a well defined communication pattern for problem identification and solution. It assists the end user by guaranteeing that someone will be available to solve a problem. The data processing staff gained a means to identify common problems and to ensure that the proper person was assigned to the problem. The Help Desk has also provided a link in the career path available to departmental staff. In the past Operations Trainees could choose to remain in operations or, if they received additional training, they could move into programming. Programmer Trainees would have technical skills but little understanding of the organization. Now the Help Desk provides a branch from operations into either applications or network programming. It also provides new programmers with a background in the local environment, its hardware, software, and personalities.

ONLY THE FIRST STEPS

The move to a distributed data processing network has offered many opportunities and challenges to the Intermediate. The Help Desk concept solved one of the primary problems, that of human communications. But the new network also brought about new machine communication problems. To handle the growth of remote devices a network of remoted multiplexors was installed. To control and monitor this communications network additional staff and equipment was acquired. The incipience of local data processing brought about promise for the development of new applications. It also brought about concern for the compatibility of local and host data bases. User training materials and practices also had to be modified for the distributed network. Each of these areas has taken staff time to solve.

The Help Desk is only one of the changes that occurred in the data processing staff as a result of the shift to a distributed system. Many others are evolving as additional users and applications adopt a distributed philosophy. The move has opened new applications and has increased the efficiency of the Intermediate's hardware. The organization and its staff have also had to make changes to achieve the benefits of distributed processing. While some of these changes were known and expected others were not. The evolving network demands a staff and organization that is resiliant and is eager to change to best serve its clients.

INSTRUCTIONAL COMPUTING IN TEXAS PUBLIC SCHOOLS - APPLICATIONS AND ADMINISTRATION

Judy Bandy Mondy
East Texas State University

INTRODUCTION

School administrators have faced increasing public demands in recent years. The American people have come to expect today's administrators to be effective decision-makers, leaders in their communities and in the field of education, far-sighted planners, and resource managers for their school districts. Administrators, however, have often found the environment in which they work plagued with uncertainty and their working conditions characterized frequently by inaccurate information and conflicting societal pressures. Schools administrators have been further challenged to make decisions regarding new technological advances such as computer systems that involve large financial outlays of local tax dollars. This decision-making process has been further confused by conflicting claims regarding computer capabilities and terminology (1).

Computer technology is evidenced today in areas ranging from health care to automotive design. Computers have become more prevalent in homes as hobbies and in toys; people have used computers to compose music, play electronic games, draw pictures, and write poetry. Not only have the lives of most Americans been affected daily by computers, but it was also reported that "high school students are increasingly choosing colleges on the basis of their computer facilities" (2). Sitkin (3) summarized the impact of computers and public attitudes toward computer integration in the following:

> Computers are already being combined with existing tools and appliances in watches, television sets, airplanes, calculators, washing machines, telephones, ovens, automobile transmission and control functions, children's games and the like. This application of electronic intelligence to everyday living is already generally accepted without blinking an eye.

Recent studies have indicated that an increasing number of school administrators have made the decision to use computers in the instructional process. Studies funded by the National Science Foundation and conducted by the American Institute for Research reported a steady increase between 1969 and 1974 in instructional use of computers in the nation's secondary schools (4). Altermott (5) surveyed elementary and secondary schools in Minnesota and found that instructional computer terminals had increased in schools from four hundred in the 1973-74 school year to eight hundred in 1975-76. Altermott also found that districts having access to instructional computer usage had increased from 78 (18 percent) in 1973-74 to 307 (70 percent) in 1975-76. In Texas, Terry (6) found that only a small percentage of school districts serving from one to five thousand students used a computer for instructional purposes. Districts that served from five to twenty-five thousand students showed a narrow increase in usage, while 79 percent of the large districts that served over twenty-five thousand students offered some form of instructional computing.

As computers have become affordable and have been recognized as an effective instructional tool, an increased number of school administrators have faced the problem of establishing an instructional computer program or evaluating and improving one already in existence. Berryman, Doring, and O'Rourke (7) found that one of the major problems school administrators had regarding computer-assisted instruction was "gaining access to programs in operation at other schools throughout the state." They further encountered little comparative data to aid in administrative decision-making. This survey attempted to provide public school administrators in Texas with comparative data regarding the application and administration of instructional computer programs.

METHODOLOGY

Study data were obtained from questionnaires sent to the superintendent or data processing administrator in each of the 1,102 public school districts in Texas during the fourth week of January, 1979. The survey instrument was directed toward gaining information regarding administration, instructional applications, hardware and software, and budget. Follow-up letters and additional questionnaires were mailed to nonresponding administrators approximately three weeks later. By the first week in March, 1979, questionnaires had been returned by 969 school districts (88 percent), and no additional follow-ups were made.

The returned questionnaires were classified into four groups reflecting district average daily attendance (ADA) for analysis. Group 1 consisted of 710 school districts (ADA less than one thousand students). Group 2 consisted of 301 districts (ADA of one thousand to five thousand students), and Group 3 consisted of 74 districts (ADA of five thousand to twenty-five thousand students). Seventeen districts comprised Group 4 (ADA greater than twenty-five thousand students. Returned questionnaires totaled 613 (86 percent) from Group 1, 274 (91 percent) from Group 2, 67 (91 percent) from Group 3, and 15 (88 percent) from Group 4.

FINDINGS

Findings are presented in four divisions. These divisions include administration, instructional applications, hardware and software, and instructional computer budget allocations.

Administration

A total of 129 responding districts (13 percent) reported computer usage in the instructional program. Twenty-five districts (4 percent) in Group 1 and twenty districts (20 percent) in Group 2 cited instructional computing activities. A total of thirty-six districts (54 percent) in Group 3 reported instructional computing as did thirteen districts (87 percent) in Group 4.

Data obtained from the survey indicated that the majority of the reporting districts (64 percent) involved between one and two hundred students in the instructional computing program. The remaining districts either reported other ranges or failed to respond to the inquiry. Twenty-one districts (84 percent) in Group 1, thirty-seven districts (67 percent) in Group 2, and twenty-one districts (58 percent) in Group 3 recorded that between one and two hundred students participated in instructional computing activities. In Group 4, however, five districts (38 percent) reported serving over one thousand students, and three districts (23 percent) involved between 201 and 400 students in the instructional computer program.

The data indicated that 97 of the 129 districts (75 percent) cited that between one and ten teachers used the computer instructionally. The other districts offered varied responses. Twenty-three districts (92 percent) in Group 1 reported between one and ten teachers involved in computing activities. Forty-two districts (76 percent) in Group 2 and twenty-seven districts (75 percent) in Group 3 listed that between one and ten teachers used computers. Six districts (46 percent) in Group 4 stated that more than forty teachers were involved in computing activities, while five districts (38 percent) reported usage by between one and ten teachers.

The position title listed most frequently as the employee directly responsible for the administration of the instructional computer program was teacher (40 percent). The majority of the reporting districts in Group 1 (68 percent) listed teacher as did half of the responding districts in Group 3. While Group 2 districts offered the widest range of titles, the position named most often was teacher (29 percent). Group 4 districts reported that the responsible employee held one of the following titles: assistant superintendent (15 percent), director/other (15 percent), coordinator (15 percent), and consultant (15 percent).

The data also indicated the position title of the person to whom the instructional computing administrator most often reported. These titles included principal (36 percent), superintendent (28 percent), and assistant superintendent (26 percent). Group 1 districts most frequently cited principal (44 percent) and superintendent (36 percent), while districts in Group 2 reported principal (47 percent) and superintendent (36 percent). Responding Group 3 districts most frequently named assistant superintendent (31 percent) and principal (25 percent), and Group 4 districts listed assistant superintendent (38 percent) and director/other (25 percent).

Fifty-six of the 129 responding districts (43 percent) reported employing between one and three persons, other than classroom teachers, to support the instructional computer program. The remainder of the districts either did not employ additional personnel, did not respond to the question, or indicated other ranges. Between one and three support personnel were reported by eleven districts (44 percent) in Group 1, twenty-six districts (47 percent) in Group 2, thirteen districts (47 percent) in Group 3, and six districts in Group 4 (46 percent).

Computer-related in-service education was offered by 112 (87 percent) of the reporting districts, while fifteen districts (12 percent) offered no such program, and two districts (2 percent) did not respond to this question. Twenty districts (80 percent) in Group 1 offered computer-related in-service education as did forty-seven districts (75 percent) in Group 2. Thirty-three districts (92 percent) in Group 3 and twelve districts (92 percent) in Group 4 also offered in-service education for involved teachers.

Development of evaluation programs to assess the instructional computer program was reported by forty-eight (32 percent) of the 129 districts. Eighty districts (62 percent) indicated they had not developed an evaluation program. Evaluation programs had been developed by six districts (24 percent) in Group 1, twenty-two districts (40 percent) in Group 2, thirteen districts (36 percent) in Group 3, and seven districts (54 percent) in Group 4.

Instructional Applications

Data obtained in the survey indicated that computer usage was most frequently recorded in the following areas: computer-assisted instruction

(63 percent), as a computational aid (60 percent), and counseling and guidance (50 percent). Group 1 districts most often reported computer-assisted instruction (CAI) and computation. Similarly, districts in Group 2 named CAI and computation, while reporting districts in Group 3 listed CAI, computation, and teaching data processing most often. Districts in Group 4 most frequently cited computation, CAI, and teaching data processing.

Computer usage in elementary subject areas was found to be most prevalent in the areas of mathematics and reading. Usage in secondary subject areas centered primarily in mathematics and teaching BASIC.

Of the eighty-one districts reporting CAI, the three most prevalent techniques used in CAI courseware were drill and practice (91 percent), tutorial (65 percent), and problem-solving (60 percent). The major source of courseware used by these districts was Educational Service Center personnel (63 percent).

Hardware and Software

Although the data indicated that a variety of hardware were used in instructional computing systems, many administrators did not complete this section of the questionnaire. The three most frequently cited commercial sources of CPU's were Computer Curriculum Corporation, Hewlett-Packard, and Tandy Corporation. The three vendors that had the greatest number of CPU's in instructional systems were Tandy Corporation, Apple Computer, and Computer Curriculum Corporation.

Statistics concerning the number of students served in the instructional computing program and the number of interactive terminals available for student use were provided by 113 districts. A student-to-terminal ratio of 42:1 was calculated from a total of 58,417 students served and 1,390 available terminals. Group 1 districts had a 42:1 ratio, while districts in Group 2 had a 28:1 ratio. Districts in Group 3 had a ratio of 31:1, and Group 4 districts had a ratio of 57:1.

The source of hardware most frequently listed by responding districts was Educational Service Centers (50 percent). However, many districts cited access through more than one source. Group 1 districts most often acquired hardware from an Educational Service Center (56 percent) as did Group 2 districts (55 percent). The majority of Group 3 districts (53 percent) owned or were purchasing their own hardware as were most Group 4 districts (67 percent).

The responding districts recorded a total of 386 locations at which students could interact with the computer, or a mean location of 3.0. Group 1 districts recorded twenty-five sites for a mean location of 1.0. Districts in Group 2 offered 107 locations, producing a mean location of 1.9; and Group 3 districts indicated 108 sites, which produced a mean location of 3.0. Responding districts in Group 4 reported 146 locations for a mean of 11.2.

The data indicated that the largest number of reporting districts (56 percent) provided thirty-one to forty hours of computer time for instructional use each week. Fifteen districts (60 percent) in Group 1 and twenty-nine districts (53 percent) in Group 2 maintained thirty-one to forty hours per week for student use. In Group 3, twenty districts (56 percent) tallied thirty-one to forty hours per week as did eight districts (62 percent) in Group 4.

The predominant programming language used in instructional computer systems was BASIC (86 percent). BASIC was named by twenty-three districts (92 percent) in Group 1, forty-seven districts (85 percent) in Group 2, thirty-one districts (86 percent) in Group 3, and ten districts (77 percent) in Group 4.

Instructional Computer Budget Allocations

Of the 129 reporting districts, sixty-five provided all the data needed for the budget analysis. These districts cited a total instructional budget of $566,407,076, a total computer budget of $4,328,233, and a total instructional computer budget of $1,312,018. The total instructional computer budget represented 30 percent of the total computer budget. In Group 1 and Group 2 districts, the instructional computer budgets represented 67 percent of the total computer budgets, while the instructional computer budgets for districts in Group 3 represented 38 percent of the computer budgets. The instructional computer budgets for districts in Group 4 accounted for 12 percent of the computer budgets.

The data reflected that the instructional computer budgets of reporting districts represented .2 percent of the districts' instructional budgets. Instructional computer budgets in Group 1 districts accounted for .8 percent of their instructional budgets, while they represented .5 percent of the instructional budgets in the Group 2 districts. Instructional computer budgets in Group 3 districts comprised .2 percent of the districts' instructional budgets, and in Group 4 districts, instructional computer budgets represented .1 percent of the total instructional budgets.

An average cost of $65.58 for each student served was found for the sixty-five districts. A cost of $69.83 per student served was found for Group 1 districts, and a figure of $55.12 per student served was determined for Group 2 districts. The cost per student served was $85.89 for Group 3 districts and $52.42 for Group 4 districts.

CONCLUSIONS

Analysis of the data presented in the study seemed to justify the following conclusions:

1. While the overall percentage of instructional computer usage in public schools was relatively small (13 percent), instructional computing seemed to have impacted public education in Texas. The utilization of instructional computing appeared to increase appreciably with

district size; over half of the reporting districts in Group 3 recorded instructional computing as did almost 90 percent of the responding districts in Group 4. While only 4 percent of the Group 1 districts reported instructional computing activities, several of the districts that cited usage were very small. For example, two districts reporting instructional computer programs had an ADA of 114 each. Thus, district size may not have been the primary determinant of instructional computer applications.

2. A recursive model or profile of a typical instructional computer system did not seem to emerge from this study. Instructional computer systems appeared to vary greatly within the same ADA group as to responsible and reporting administrators, hardware, sources of courseware and hardware, and budgets. Although the titles of the responsible and reporting administrators varied, program control in Group 1 and Group 2 districts tended to be at the building level, while districts in Group 4 seemed to favor central office control. Districts in Group 3 appeared to be split between building and central office control.

3. Regional Educational Service Centers appeared to have performed an important role in furthering instructional computing by providing needed sources of hardware, courseware, and inservice education programs. This technical assistance, not always available in smaller school districts, may have enabled districts to offer instructional computer services that they might not have been able to do so otherwise.

4. The proliferation of minicomputers appeared to have impacted the use of computers in public education in Texas. Districts in all four ADA groups reported the use of these systems, but Group 1 districts almost exclusively cited utilization of the relatively inexpensive systems. Since most minicomputers utilize the BASIC programming language, the proliferation of these smaller systems may have been one important reason why BASIC was reported almost exclusively as the predominant language used in instructional computer systems.

5. While most districts which offered instructional computing provided related inservice education programs to district teachers, few districts had developed evaluation programs to assess the effectiveness of these systems. This lack of evaluation may have reflected an embryonic stage of development for instructional computing in which the program had not yet been recognized as a formal segment of the instructional program. Also, since financial resources for instructional computer systems were primarily local funds, the question of evaluation was left to the individual district.

REFERENCES

(1) American Association of School Administrators Committee on Technology and Instruction. 1970. Instructional Technology and the School Administrator. Washington: AASA.

(2) "Living: Pushbutton Power." February 20, 1978. TIME, pp. 46-49.

(3) Sitkin, Irwin J. December, 1978. "The Intelligent Putter," Data Management, 16:46.

(4) Bukoski, William J., and Arthur L. Korotkin. September, 1975. Computing Activities in Secondary Education: Final Report. U.S., Educational Resources Information Center, ERIC Document ED 112 935.

(5) Association for Educational Data Systems. 1976. Today's Revolution: Computers in Education. White Plains, N.Y.: International Business Machines Corp.

(6) Terry, James P. 1976. "A Survey of Electronic Data Processing Usage in the Public School Districts in the State of Texas." Unpublished Doctoral dissertation, East Texas State University.

(7) Berryman, John, Richard Doring, and David O'Rourke. March, 1976. The Use of Computers for Instructional Purposes--A Preliminary Survey of Selected Schools in Illinois. Illinois Series on Educational Applications of Computers, Number 4. U.S., Educational Resources Information Center, ERIC Document ED 138 281.

THE COURSEWARE DEVELOPMENTAL CYCLE AS A MODEL FOR PLANNING AND IMPLEMENTATING AUTOMATED TRAINING SYSTEMS FOR AN ON-LINE CLAIMS PROCESSING SYSTEM

Norman Orr/Gary Zaccagnini
Veterans Administration

The Veterans Administration has developed a nationwide on-line claims processing system. The system has about 2,600 video display terminals (VDT's) in 58 locations. In each location (regional office) there are three or four different divisions or operating elements which use the VDT's. Initially the implementation was planned within a six week period nationwide.

The trainees who used the system ranged from beginning employees to those whose careers were almost finished (30 years of service). They ranged in age from 18 to 70 years of age. Some had not completed high school; others had graduate degrees. Many had little or no familiarity with electronic equipment or a typing keyboard (the VDT has an associated keyboard similar to a typing keyboard).

The training responsibility was limited in scope. The training course development group assumed that the trainees had substantive knowledge of the job they currently performed. The training responsibility was limited to training the employees to use the VDT to perform the job, instead of doing the job with a paper medium.

The application which was developed consisted of a series of screens required to process a claim or inquire against a data base for information retrieval. The training job was visually oriented, but required a combination cognitive/psychomotor activity.

Early manual training efforts, in a pilot version of the proposed system, revealed that hands-on experience was an absolute necessity for an effective learning experience.

Since adequate computer power was available in the proposed system, and since each regional office would have VDT's a decision was made to use computer assisted instruction (CAI) as the chief medium for training employees to use the system. However, that requirement was not specified in the request for proposal (RFP) which was issued for the system. When the contract was awarded, the problem soon became, "How do we make CAI available for the system?"

The winning vendor had no off-the-shelf CAI software. Early efforts to locate a transportable, or easily modifiable, CAI software package were unsuccessful. Our systems analyst concluded that the most efficient, and possibly only, way to provide CAI for the system was to develop a software package uniquely for the system.

In the interim, based on our experience in the Pilot, we developed a courseware development cycle tailored to the specific training task. A clearly understood instructional objective is always necessary, but given our limited training responsiblity, it was even more important. Our final product not only had to be clearly understood in intent by the trainee, but also had to have the concurrence of the user departments for which the training material was being prepared. Therefore, the beginning of the courseware cycle, was writing a clear, concise, instructional objective. This objective was placed on the first page of the instructional material. Because of the wide variation in age, educational background and work experience, a population analysis was our next step. The population analysis was different for different groups of courseware, since not all employees would take all parts of the total courseware package.

The population analysis was followed by a task analysis. The application required a logical, step by step approach as did the sequencing of screen formats to perform the tasks. It was important that each step be included and in the proper order. The task analysis assured us that we had met this objective in the courseware development cycle.

Although not all employees took all sections of the courseware, everyone who was trained in any function needed orientation to the system and training in keyboard operation. The simpler applications also served as a basis for the more complex functions. Therefore, we could assume a core of knowledge for the more advanced courseware which served as a base for the more advanced functions.

CAI itself provided the hands-on experience necessary; however, the trainees were expected to begin work as soon as possible after the training was completed. Both we, and they, had to know that they were capable of performing the function. So each section of the courseware contained a criterion test. Without successful completion of the test, the trainee was not permitted to perform the task in production.

When we began to develop the necessary CAI software for the system, we discovered that this same courseware developmental cycle should serve as a model for software development. The beginning of any adequate systems design is a clear statement of the purpose of the system. The instructional objectives served as an overall, detailed outline of the specific routines one would have to be able to accomplish to effectively learn the production tasks for which training was required. Therefore those objectives, collectively, served as a set of requirements for the CAI courseware, and formed the outline for a user request document.

Further, the task analyses served as detailed guidelines for development of the detailed functional design document. All the essential steps were there, and they needed only to be restated and filled out in proper documentation format. These two pieces of documentation served well as the source from which detailed technical specifications could be prepared.

Although we did not do so, the criterion tests could serve equally well as the basis for the audit routines if certification of the system is required. The tests effectively prove that the trainee is able to perform the entire functions correctly. The examples in the test could serve as test data and the system's successful execution of the functions of the tests would prove the effectiveness of the software.

The only part of a traditional developmental cycle which is not covered in the outline above is a feasibility study. The population analysis can serve as the basis of this feasibility study. The population analysis will not supply all the data necessary for the feasibility study. The study will require some comparisons of traditional or existing training methods versus the automated training system being developed. Any intangible gains will also have to be considered. But the population analysis will point out many aspects of the training task which would not otherwise be obvious and provide the basis on which the feasibility study could be done.

Not all applications will require this same courseware developmental cycle, but all applications will require some courseware developmental cycle. And some elements of this development cycle should be included in any courseware development cycle. Whatever courseware developmental cycle is used, if it is well designed, can serve as an effective basis for a developmental cycle for an automated training system.

Sometimes the situation is reversed. One has the responsibility to develop an automated training system and subsequently a courseware development cycle. The reverse should be true. If one has a well defined systems developmental cycle that cycle should serve as a basis for the development of a courseware development cycle.

This mutuality of systems and courseware developmental cycles is especially true in an on-line system, which uses video display devices. VDT systems are the wave of the future. The effective recognition of this mutuality will cut developmental costs and time for the development and implementation of complete automated training systems and the courseware for those systems.

EMPLOYING DATA-MANAGEMENT PACKAGES TO MODERNIZE YOUR PLANNING DEPARTMENT

Neil G. Pedersen
Richmond Public Schools

The history of administrative data processing in the public school sector is not filled with unparalleled success stories. Although a few school systems have been fortunate enough to uncover a budding "computer freak" teaching in the local high school and has built a computer center around him, in general, school systems lag far behind the private sector in developing economical, efficient, and technically innovative data processing systems. In this paper, an attempt is made to paint a picture of the data processing problems which may be facing thousands of school administrators across the nation and then to present one school system's strategy for overcoming some of these obstacles.

State of Data Processing in Educational Administration

After reviewing literature on educational data processing, it was determined that no comprehensive national survey has been conducted recently to assess the environment in which computer centers operate in our schools. In the absence of any empirical data, the following generalizations are proposed to describe the state of data processing in most public school systems.

1. Small school systems (under 10,000 students) make little if any use of data processing in their administrative practices despite the advent of mini-computers which make data processing affordable in such settings. Nor do these school districts take advantage of teleprocessing capabilities unless an intermediate school district or other educational institution makes its services readily available.

2. Most educational administrators are still oriented toward collecting and analyzing data in a basically manual fashion. The electric calculator rather than the computer, is the basic tool of most administrators.

3. Data files are generally structured and used for only one application and are seldomly integrated into a management information system.

4. School systems are losing the battle with private industry for competent data processing staff. Staff turnover and inadequate training often thwart efforts to improve data processing systems.

5. Many data processing directors are not trained in this field and, consequently, have a limited perspective of computer technology.

6. Many school systems mistakenly purchased hardware several years ago and, consequently, are now wedded to obsolete systems.

7. When school systems are required to use the local governing body's computer facilities, the arrangement is often perceived as unsatisfactory to all concerned.

8. Because most educators have a limited knowledge of data processing, they have difficulty communicating their needs to data processing staff who speak a different jargon.

Local Conditions and Needs

Many of these same problems were faced by the Richmond, Virginia Public Schools in the period between 1974 and 1977. Its data processing department was incorporated into the City's in 1974 such that the City assumed responsibility for meeting all of the hardware and software needs of the Schools. The Schools paid the City approximately $500,000 annually to cover personnel, machinery, and overhead costs. Generally speaking, in order to effect system modifications or the development of new applications, a work request normally had to be submitted through the Schools' Director of Planning and Development to the City's Director of Data Processing. The City then determined the feasibility of the request and if it was approved, assigned a systems analyst and provided a cost estimate.

Such a system caused a number of problems. Turnaround time generally varied anywhere from a few weeks to a couple of years. Frequently, lack of communication led to the production of reports which were either inaccurate or incorrectly formatted. Requests for information needed within 48 hours could not be programmed within that time frame and, furthermore, because the City programmed exclusively in COBOL, it was seldom economically justifiable to write a COBOL program which might be used only one time.

Consequently, the City Data Processing Department was relied upon to develop and maintain basic systems; attendance, grade reporting, course scheduling, payroll, and financial accounting. Requests for reports which were needed immediately, which involved more than basic statistical analysis or which would be generated infrequently were processed on a manual basis because of the constraints indicated above. Frequently the cry was heard from the schools, "You've got this data in the computer. Why do you have to keep asking us for the same information?" Clearly, the School's administrative support systems were not adequately addressing the information needs of the schools or the central office administration.

The Department of Planning and Development was charged with the responsibility to coordinate data processing activities with the City, conduct major surveys and coordinate the processing of other reports, and to provide information to all segments of the school community. In 1977 the Department was composed of nine administrators, five technical support personnel, and six secretaries. All of the administrative staff had risen up through the teaching ranks and only one member had any experience with computer programming. Because of the problems associated with data processing described previously and the staff's unfamiliarity with data processing, virtually all requests for information were processed manually.

What was needed was an automated system which could: (1) provide reasonable turnaround time on requests for information - often within a 24-hour period; (2) retrieve data from computer files associated with the major systems and generate different analyses of these data; (2) reduce the number of requests made of schools to provide information which was already "in the computer;" (4) facilitate the analyses of large amounts of data with several different types of breakdowns - something which was not feasible with calculators; and (5) accomplish these objectives with no additional staff and at a reasonable cost.

Solution

It was evident that the answer to this problem was not simply to send more work requests to City Data Processing. The time and associated expense required to write COBOL programs conflicted with the requirements described above; furthermore, competition from the private sector had reduced the City's staff to a skeletal level which could do little more than maintain the existing systems.

As a first step, the Statistical Analysis System (SAS) software package was leased from the SAS Institute at a very modest cost. One of the Research Specialists in the Department of Planning and Development taught himself the language and began to write a few SAS programs. Soon after this, a vacancy in the department was filled with someone with an educational research background who had some experience in writing FORTRAN programs and using the Statistical Package for the Social Sciences (SPSS). He, too, learned SAS through self-instruction.

Quickly the word began to spread that analyses which had been requested for years could now be generated with relative ease. As the requests for ad hoc reports began to pour in, it quickly became evident that two administrators could never meet the needs of the school system. What was needed was an entire revamping of the Department of Planning and Development: one which would change the orientation of the 20 staff members from the electric calculator to the computer. In two short years this essentially has been accomplished.

In order to effect this transformation it was determined that a data-management software package would have to be found to meet the needs of the users, additional staff would have to learn how to use the package, and some remote computer hardware would be necessary to make the City's computer more accessible. Fortunately, as more experience was gained in writing SAS programs, it was apparent that this software package had almost unlimited potential. Several different files can be merged, concatenated, or sorted almost effortlessly, and reports can be generated in a predetermined format or one specified by the programmer. The high quality of this software package is supported by the fact that it is installed in over 800 sites across the United States.

In addressing the question of in-service, a decision was made to conduct classes within the Department using staff who had previously learned SAS to instruct others. Of course, most of the learning took place as the staff began to write its own programs and encountered difficulties. In 1978 two Research Assistants (formerly English and Social Studies teachers) began using SAS to manage data which had previously been collected and analyzed manually. The fact that a computer background is not essential in order to learn SAS was demonstrated in that neither of these people had ever run any type of computer program. In the fall of 1979, the in-service program was revitalized and included all administrators and technical assistants in the Department. Secretaries received in-service relating to remote data entry.

In addition to the software and in-service needs, it was desirable to acquire remote processing equipment in order to make the computer more visible as well as more accessible. Three CRTs were leased in 1979 and one printer is budgeted for 1980. The terminals are used in conjunction with ROSCOE to enter programs and data and to determine job status on the main computer.

Results

Today, every administrator in the Department has at least minimal skills in using the SAS package and the entire staff can use the CRTs. Four positions are devoted almost exclusively to the processing of data with SAS, and it should be mentioned that the size of the department staff has been reduced by two positions. But most importantly, virtually everyone in the department has changed his approach to information management. When an assignment is received, the first instinct is to determine how the computer can assist in the task. While the computer has not completely replaced the calculator, it has surpassed it in priority in the eyes of the Department staff.

A summary of just a few of the projects in which SAS has been used is provided below. In reviewing this list, it should be kept in mind that most of these tasks represent expansion, not replacement, of Department responsibilities.

1. Geo-Based Data System. Prior to 1978, "spot maps" were constructed for each grade level by manually marking on a set of large maps the location of every public school student in the City of Richmond. This required the employment of approximately four persons for two months. Today the same information is generated automatically by using the U.S. Census DIME file, associated software, and the SAS package.

2. Student Population Projections. The above system provides the foundation for computerizing the process of projecting student populations in the years ahead. This procedure is now essentially totally computerized and consequently employs a much more sophisticated model than was used when projections were made manually.

3. Survey of Students for Impact Aid (PL 81-874). Again, the geo-based data system provides a starting point for a system which automatically locates all students who reside in federally subsidized housing projects. Names of these students are printed and forwarded to schools: labels are printed and affixed on the survey cards.

4. Operating Budget. A system of SAS programs (developed by a local consultant) is used to compile, maintain, and analyze appxoimately 5,000 entries in the expense portion of the general fund operating budget. Reports are generated by function, program, and location. Data are maintained by the budget assistant using a ROSCOE terminal. Turnaround time has been reduced greatly and the accuracy of the data is virtually 100 percent whereas it was unacceptable when farmed out to keypunchers in the City Data Processing Department.

5. School Census. Every three years school divisions in Virginia must conduct a census of all school-age children in the community. In 1980, this entire system has been programmed with SAS.

6. Records Library. Thousands of reports which either emanate from or flow through the Department of Planning and Development are cataloged according to subject, author, title, type, and date. The SAS package is used to locate reports and generate periodic as well as ad hoc listings of the collection.

7. Design of Feeder System. In 1979 the Richmond City Schools implemented a feeder system for its elementary, middle, and high schools. SAS programs generated student demographic data used in determining how students would feed and in the drawing of new attendance boundaries.

8. Personnel Evaluations. Personnel evaluations are keyed-to-disk and SAS programs summarize and analyze the results annually for five different evaluation forms.

9. Gifted and Talented Students. SAS programs were written to create historical test score files and then to identify students who met specified criteria. Other programs maintain a longitudinal file of gifted and talented students identified through this and other procedures. Schools annually receive an updated list of such students at the beginning of the school year.

10. Special Education Students. SAS programs facilitate the completion of several federal and state reports on special education students.

11. Course Code Book. Course descriptions and specifications are entered on ROSCOE terminals by Department secretaries, and the Course Code Book is computer printed using SAS. Computerization facilitates making revisions and also provides a master course file used in student scheduling.

12. Test Scoring. Programs are written to score high school graduation competency tests rather than having them scored commercially. This reduces turnaround time and saves approximately $5,000 annually.

13. Instructional Management Systems. The Communicative Arts Program in grades 6-12 relies upon SAS programs to generate systemwide, school, class, and individual reports on student progress through the curriculum.

14. Miscellaneous Requests for Information. Approximately 100 programs are written annually to answer questions raised by school personnel, research agencies, and members of the community.

When one considers that most of these systems were developed by a staff with virtually no data processing background within about a two-year period, the record is one of which the Richmond Public Schools can be proud.

Summary

The Department of Planning and Development in the Richmond Public School System was charged with the responsibility of providing information to a variety of audiences in a timely and efficient manner. The City's Data Processing Department was of limited use because of its COBOL orientation and its inability to attract and retain a sufficient staff. Consequently, the Department leased the Statistical Analysis System (SAS) and had it installed on the City's computer. Over a two-year period all members of the Department of Planning and Development were given in-service on data processing fundamentals, SAS programming, and remote data-entry. The SAS package is now used by many members of the Department on a regular basis as a tool for generating information for school administrators and instructional staff.

This case history is provided as a model to school systems who are dissatisfied with aspects of their data processing services or who do not even now use a computer in their generation. It illustrates one strategy which school systems can use to provide vastly more information in less time and for less money than is possible using desk-top calculators or conventional computer programming techniques. In light of increased demand and, consequently, cost for data processing personnel, the Richmond Public Schools believes that this method of providing timely information will become increasingly popular in educational institutions in the years ahead.

More Hands for Teachers:
Report of an
Instructional Computing Study
For the State of Florida

M. D. Roblyer
Florida Department of Education

In 1977, Florida initiated a plan for equitably distributing *administrative* computing resources throughout the educational system. However, *instructional* computing was not addressed in this plan. In the 1979 session, the Florida Legislature directed the Department of Education to study the implications of instructional computing for Florida public education. The results of this study, and the recommendations submitted by the Commissioner's Advisory Committee on Instructional Computing, may be seen as the first step in establishing a coherent policy and plan for facilitating instructional computing in the Florida educational system.

Introduction

The 1970's witnessed several giant steps toward the availability of powerful, inexpensive computing capability which could make technology practical, as well as useful, in the classroom. Yet despite this capability and increasing evidence of its use in educational settings, federal funding for instructional computing activities has actually decreased during the decade, and state-level planning efforts for the area are still few and fragmented. The result is a large area of educational technology whose growth and development have been largely undirected and unevaluated. As was emphasized in a federal study of computers in education (U. S. House of Representatives, 1978), what is lacking is a "coherent, rational policy towards the use of computers for improved learning in our society." Well-planned policy is an essential component in assuring that optimal advantage is taken of this powerful resource.

In an effort to define such policy for the State of Florida, the Florida Department of Education was recently tasked by the state legislature with studying the "role, feasibility, and cost-effectiveness" of instructional computing for public education. The findings and recommendations of the study, as well as the strategies employed to carry out the effort, are reported for the benefit of others involved in planning for instructional computing programs.

Background. The 1979 law requiring the study was welcomed by those in the educational community either already involved in instructional computing or anxious to begin such involvement. Recently-developed computer-based products, especially in the microcomputer area, seem to have captured the imagination of both administrative and teaching personnel as never before. However, there was little state-level attention to the area, either in funding or in providing guidance on the selection and implementation of computer-based instructional products, until the study was undertaken. While some members were able to proceed with local resources and expertise, many felt the need for more technical assistance than was available in their organizations, before they invested in computer methods and materials.

There was growing acknowledgement that the impact of computers on education could not be ignored. Many perceived that the availability of inexpensive powerful computers in society has created new problems for education, as well as promised new ways of addressing existing problems.

Already evident is the perception that "computer literacy" must become a new basic skill for education, both to allow student use of computerized methods for information retrieval and problem-solving, and to provide equal opportunity for the new jobs created by computer technology. This new curricular need, in addition to the need for ways of using technology to make teaching methods more productive, made it imperative that the state begin planning for system-wide use of instructional computing.

The need for state-level attention to instructional computing, however, only became widely recognized as a result of another, related project funded by the Florida legislature. In 1977 a law was passsed which addressed the need for more equitable distribution of educational computing resources throughout the State. Funds were allocated to promote the sharing of existing computing resources among members of the system. However, these funds were to address primarily administrative activities rather than instructional ones.

A primary strategy of the equitable distribution effort was to provide guidance on purchase and use of computing resources in order to obtain optimal benefit from educational computing funds. It became apparent that many purchases and substantial use was being made of computer resources in the instructional, as well as the administrative, areas of the educational system. Since these activities represented a significant state-wide expenditure, the need for technical assistance in instructional computing areas became recognized. However, two problems remained.

First, there was a prevailing assumption, among system members and state-level personnel, that the same kinds of technical assistance methods to address administrative computing activities would suffice for instructional applications. Since both were included under the umbrella heading of "educational computing", there was a prevalent feeling that instructional computing was merely "data processing at the classroom level." Some effort was required to change this perspective and to delineate the crucial differences which must bear heavily on technical assistance strategies used to support the two areas.

As the information in Table 1 indicates, not only are the applications in the instructional area much different in purpose from administrative applications, the expertise required for each area also differs significantly. Recognizing the distinction between the computer as information processor and the computer as instructional medium seems an important step toward the development of policy and procedures for instructional computing. And yet, separating the activities in this way presented a further problem, this one concerned with finding state-level support for non-administrative computing.

Perhaps it should not be surprising that it is easier for many educational personnel to accept the necessity for computers in the school district office than it is for them to recognize the need for computers in the classroom. There is probably as much data on the time-savings achieved through computer-based learning methods as there is on the efficiency of some computerized administrative applications. However, the computer as innovative teaching method has several negative connotations not attached to the computer as record-keeping tool.

Much was learned from the instructional computing projects of the 1960's and early 1970's, and the image of computer-based methods which developed during this period is still very vivid in the minds of many educators. That image is generally less than favorable. The powerful resource which was to revolutionize education by solving the majority of its problems was often used as an expensive electronic page-turner. Instructional materials were often developed by technicians with little expertise in principles of instruction and learning. The resulting perception of computer-based methods was effectively summarized by a Florida legislator who pointed out that "we already tried those innovative methods in the 1960's and found out that they don't work!"

Clearly, the image of instructional computing needed changing. Whereas before, computer use in education was equated with a kind of "panacea that failed," it needed to be viewed as one of many instructional tools which could, with proper planning, effectively address a few key problem areas. Defining these areas, and identifying specific ways in which computerized methods could be applied, became the focus of the state-level planning study.

Study Procedures

Participants. Although the study was performed under the direction of Department of Education staff, it seems imperative in such efforts that members of the educational system be closely involved with planning for activities which could impact their organizations in significant ways. For this reason, the Commissioner of Education appointed an ad hoc Advisory Committee on Instructional Computing to carry out study activities. Since the focus of the study was to be primarily on educational issues, and only secondarily on technical ones, the 12 members were selected for their current involvement in teaching or educational administration. Several members of the Committee had, however, been involved in educational computing in some way. Some background on Committee membership is shown in Table 2.

Table 1

A Comparison of Instructional and Administrative Computing Applications

	Type Applications	Purpose of Use	Expertise Required to Develop Applications
Instructional Computing	Computer-assisted instruction Computer-managed instruction Computer-aided guidance Computer literacy	Direct aid to teaching/learning process	Instructional design principles Learning theory principles Content knowledge(i.e.math, history, reading, etc.) Technical skills
Administrative Computing	Staff payroll Staff records Student records Student scheduling Inventory	Support for clerical and record-keeping tasks of educational organizations	Knowledge of activity (accounting,scheduling,etc.) Knowledge of school administration Technical skills

Table 2

Members of Commissioner's Advisory Committee on Instructional Computing

School Districts: 6 Members

- Math Teacher
- Science Teacher
- District Superintendent
- School Principal
- Director of Evaluation
- Director of District Vocational Education

Community Colleges: 3 Members

- DP Instructor
- College Vice-President
- Director of Staff and Program Development

Universities: 3 Members

- Professor of Education
- Professor of Education
- Chairperson, Department of Data Processing

Study questions. The activities of the study were intended to address three major areas:

(1) How should instructional computing be used? A basic assumption in the study was that it is neither practical nor desirable to computerize everything which can be computerized within the instructional process. The intent was rather (a) to identify key educational problem areas to which the unique capabilities of the computer have been successfully applied, and (b) to define those areas of computer application which have been instrumental in that success.

(2) How is instructional computing currently being used? The concern in this area was to determine what has already been accomplished, both in Florida and elsewhere, toward implementing the desired activities found under question #1 above. The Committee was specifically interested in: methods and materials in current use, problems and issues involved in implementing instructional computing projects, and developing trends in both instructional computing products and methods.

(3) What needs to be done in order to make best use of instructional computing in the future? A final step in planning was to be a synthesis of all available information on past and present instructional computing activities in order to make recommendations on policies, activities, and resources required for the future for the optimal use of technological capability.

Data and information sources. In order to obtain a comprehensive summary of responses to the study questions from every relevant persective, several kinds of resources were consulted. These included:

(1) Review of literature - This resource was used primarily to address the first question concerning evidence on specific ways that instructional computing has been found to enhance teaching and learning. Both experimental studies and evaluations of projects were reviewed. The focus was primarily on recent (1975-1979) published reports. However, several previous reviews of the literature on studies performed prior to 1975 were also consulted, as were several unpublished reports available to the Committee.

(2) Surveys of current instructional computing activities - Several methods were used to ascertain the state-of-the-art in instructional computing.

- State-level Activities Outside Florida - A letter requesting information on similar instructional computing studies or other related activities in other states was sent to the chief educational administrator in each state outside Florida.
- Current Florida Projects - Several CAI, CMI, and computerized guidance systems currently in use in Florida school districts, colleges, and universities were identified. The project directors were asked to submit a written description of their activities in response to a set of questions (name of system, grade levels served, implementation problems, etc.) The directors of a cross section of these projects were also asked to give verbal testimony to the Committee.
- Florida Hardware/Software Survey - A telephone survey was performed of the 67 school districts to ascertain how many hardware units and what software packages were in place to serve students in various grade levels and content areas.
- Florida Teacher Education Survey - The public and private four-year institutions which have teacher training programs were contacted to determine what kind of computer literacy component, if any, was required as part of the program.

(3) Consultants - Expert advice on current and projected needs to address in this area was sought from two kinds of consultants. First, five nationally-recognized experts from various parts of the country were asked to supply written comments on major issues identified for them, and to participate in a subsequent round-table discussion with the Committee and staff, in order to elaborate on these issues. Secondly, the major computer industries were asked to send their instructional computing expert to meet with the Committee and present their views on the future market in the area and what they are doing now to prepare for that market.

Study meetings. The Committee and staff met a total of five times, each time to discuss some aspect of the area which had been previously researched and summarized in writing. These meetings were used for the important task of synthesizing the information compiled for the study through discussions and deliberations among those involved in the study. The final meeting was used to summarize what had been learned and to develop recommendations on what should be done with instructional computing for Florida public education.

Summary of Study Findings

Question #1: How should Instructional Computing be used? The results of a review of recent experimental and evaluation studies in the use of instructional computing methods were encouraging. The major variables considered were:

- Achievement and learning gains
- Time required to learn
- Attitudes of students
- Retention of learned material
- Costs of various methods

Most studies compared results obtained on these variables with CAI, CMI, or computerized guidance methods with those obtained on non-computerized methods. The computer-based methods were consistently indicated to be as good as or better than non-computer-based methods on each of the variables considered. The amount of time required to learn with CAI/CMI methods was usually lower, and motivation higher among students. Findings were especially promising on the variables for CMI methods. Most of the research was performed at university levels, but findings seemed to be consistent among all educational levels.

These findings seem to support the verbal claims of many who have used computer-based methods that students are more motivated to learn and require less teacher time than with traditional methods. The unique capabilities provided by computerized instructional systems appear to facilitate the interactive, individualized, and self-paced characteristics required for teaching many kinds of special-need target populations. It has been observed that without instructional computing methods, these students may never receive the attention they require. The special-needs populations include:

(1) Students requiring remediation in basic skills - Florida has already recognized the need for re-teaching functional literacy skills to students experiencing difficulty in this area. Computerized methods can help with this task in two ways: (a) By providing drill-and-practice to develop rule-learning skills and to enhance retention of learned skills, and (b) by supplying an automated system of tracking student progress and supplying specific individual information on areas of student weakness.

(2) Students with learning disabilities - Again, both drill-and-practice and management capabilities can help with the learning problems presented by many kinds of learning disabilities. A medium such as CAI is able to provide the constant patience, reinforcement, and stimulation required for these students.

(3) Students in low population courses - Content areas such as advanced mathematics (trigonometry, calculus) and advanced science (physics, chemistry) are often considered low-priority areas for staffing since they do not draw large numbers of students. Computer-based methods can help supply self-paced learning environments for these and other areas where teachers are often unavailable.

(4) Accelerated learners - Remedial instruction for slower learners is some-

times provided at the expense of attention to gifted learners. Since gifted students are also entitled to an appropriate educational program, and since these students are an important future resource for our society, there are currently programs underway to provide better instructional environments for the gifted. Computers can help by offering a powerful intellectual tool with which the student can explore all content areas, and by making possible self-paced methods for learners to acquire future knowledge and skills.

(5) Transfer students - There are many students who, for a number of social and personal reasons, move frequently from school to school. These students may have no learning difficulty other than simply getting access to material which they missed by transferring to a school with a different curricular structure. Self-paced methods offered by CAI and CMI again seem a logical solution to this problem.

Question #2: How is instructional computing currently being used? It was evident from surveys and consultation with experts in the field that, despite the dearth of state-level planning and direction, there are distinct trends in usage emerging in the educational community. There are also some common issues and problems which are becoming apparent. These may be summarized as follows:

- Delivery systems - The rapidity of change in computer technology has impacted instructional computing in several ways. There is currently considerable discussion about the size and capability required in a hardware configuration for instructional use. There are also concerns about the quality of instructional software for these systems. The technology seems to be moving toward smaller, increasingly powerful standalone microcomputers as instructional delivery systems. Although there are large, timesharing systems available for instruction, there seem to be more costs and logistical problems associated with implementing these than with microcomputer systems. However, since instructional personnel can easily become confused about various vendor claims for their products, they are often in need of advice and direction on what kinds of purchases to make to meet their needs. Currently, expertise to deal with these questions, as well as access to the hardware itself, is often centered in the more affluent school districts.

Perhaps the most difficult problem is how to assure an adequate amount of high-quality instructional software for various instructional content areas and target populations. It is often overlooked that the effectiveness of CAI depends almost totally on the quality of the lesson material (courseware). Although there are a number of microcomputers and timesharing terminals in schools at this time, the software resources are often scattered and of inconsistent instructional quality.

- Teacher training and computer literacy - The history of innovative programs in school systems has shown that success of the effort depends upon adequate training and preparation for the personnel who will implement the program. Teachers who are unfamiliar with and perhaps apprehensive about computer technology require a well-designed and comprehensive orientation program in both general computer literacy and in the use of specific systems to be implemented. It seems apparent that this orientation cannot be done quickly, in an after-school workshop environment. Some awareness of computer technology needs to be stimulated during teacher education programs, in addition to providing inservice training in the specific CAI/CMI materials to be used.

At this time, computer literacy is not addressed in any comprehensive way in Florida teacher education. A survey of Florida teacher training institutions reveals that few programs require computer literacy at any level as part of the skills for teaching. Clearly, more emphasis needs to be placed on this training if instructional computing applications are to be accepted into classroom use.

- Support requirements - Once the initial elements of the dissemination plan are set in motion, provision must be made for maintaining the quality and effectiveness of the activities. This is especially true in light of the changes occurring daily with technological products and services. The following kinds of support functions must be addressed:

Purchase agreements - Most vendors of computing equipment and software are willing to allow substantial discounts and will even include additional services, if they are approached from a systemwide level, rather than by individual districts or institutions. Several of these agreements have already been made for non-instructional computing products and have resulted in considerable cost savings for the State of Florida.

Technical information resource - Often, instructional personnel are interested in acquiring instructional computing products, but do not know where to begin looking for such products. They need a resource which could provide technical assistance initially and later help identify problem areas and suggest solutions during implementation. Such a requirement is especially evident at this time in smaller, less-affluent areas, which do not have technical resources of their own.

Communication channels - The current Florida Equitable Distribution Plan for administrative computing resources is based on the knowledge that many technical resources which are available in more affluent areas may be shared with neighboring, less-affluent members. The key to encouraging this sharing activity is communication among members of the system. It has been demonstrated that communication for purposes of sharing technical knowledge and products can do much to make better use of existing resources and thus make the system as a whole more productive. Evaluation activities - It is important that some efforts are made to evaluate the progress of instructional computing programs, in order to determine ways of revising them and making them more effective. Such documentation has been seen as a necessary formative component in assuring the continued success of any instructional activity.

- Costs - A review of work in the area of costs revealed that methods of conducting cost-effectiveness analysis have only recently been considered in the field of education. Furthermore, the staff reviewed over 300 references related to cost-effectiveness, and were able to identify only 32 which contained results supported by quantitative or empirical data. It appears that the delay in employing cost-effectiveness techniques and the inability to apply them appropriately stems from the fact that the requirements for conducting sound cost-effectiveness analysis in the area of instructional technology are difficult to satisfy. While CAI is usually found to be more expensive than traditional methods, and CMI is generally seen to have somewhat more cost advantages, these findings are able to advance only the most tentative picture of the cost-effectiveness of instructional computing. Since there is a continuing difficulty in identifying and measuring the outcomes of education, it appears impossible at this time to say definitively whether or not any one instructional method is more cost-effective than another in achieving these outcomes. However, positive indications of the instructional effectiveness of computer-based methods seem to justify more investment of resources and further study of this area, in order to reach more substantive conclusions.

Question #3: What needs to be done in the future? The answer to this question seems to lie in an immediate planning effort to address two factors. First, there is an urgent requirement to begin the development of a management system for (1) disseminating quality instructional computing resources on a equitable basis, (2) for providing training and technical support to facilitate effective use of the technology, and (3) ascertaining cost-effectiveness of these uses for various areas of education. Secondly, and equal in importance, is the need to develop strategies for dealing with current perceptions about instructional computing which could have impact on the acceptance of the technology. These implementation issues and concerns include:

(1) The management of change: Can instructional computing as an innovative method be successfully introduced in the present educational system?

(2) Impact on learning: Does instructional computing really make a difference?

(3) Effects on teachers: Will the role of the teacher change?

(4) Effects on students: Will instruction become impersonal?

(5) Localized vs. standardized delivery systems: Can systems be standardized in any way and still meet local needs?

(6) Availability of courseware: Where will materials come from?

(7) Computer literacy needs: How much literacy is enough?

Conclusions and Recommendations

The Committee's recommendations re-emphasized the need for a state-level policy for the use of instructional computing in public education. Several assumptions on which policy should be based were listed, the first and most emphatic of which was that Florida cannot afford not to invest immediately in technology for schools, colleges, and universities in the State. The Committee recommended that the Legislature change or adopt statutes which call for the DOE to facilitate utilization of instructional computing in several priority areas of need. These priority areas are to fall in two general categories as follows:

(1) Computers as a medium of instruction

- Basic skills instruction
- Problem solving
- Education for the handicapped
- Low enrollment courses

To assist in the dissemination and acquisition of materials necessary to support each of the above areas, the DOE should establish a clearinghouse of information on software, hardware, and other available materials, both inside Florida and elsewhere, to use in each area of need. Information from the clearinghouse should be supplied on request and through appropriate dissemination strategies identified by the Department.

(2) Computers as an object of instruction

- Computer literacy for educators
- Vocational and professional fields
- Computer literacy for students

The Committee also called for the support of this implementation through the following:

1. Providing information and consultation on available hardware and software and the use thereof.
2. Maintaining a catalog and inventory of available resources for instructional computing in Florida school districts and other educational institutions.
3. Assisting in efficient ways of securing discounts on hardware and software on behalf of school districts and institutions.
4. Technical assistance to school districts and institutions on individual problems related to instructional computing.
5. Facilitating and coordinating the evaluation of instructional computing projects.
6. Leadership and staff assistance to Department of Education staff members in relation to the priorities recommended for policy implementation.

Reference

U. S. House of Representatives Committee on Science and Technology. *Computers and the learning society report*. Washington, D. C.: U. S. Government Printing Office, 1978.

Note:

Tables, charts, and further information which could not be included here because of space limitations will be available when the paper is presented at the AEDS conference.

DIFFERENTIAL IMPACTS OF A COMPUTER INFORMATION SYSTEM ON SELECTED HUMAN SERVICE AGENCIES

CHARLES W. RYAN
ROBERT J. DRUMMOND
GERARD BILODEAU

1 The use of computer based information systems is receiving increasing acceptance in educational and human service agency environments. Public schools and post-secondary institutions have been willing to accept computer information systems such as SIGI (Chapman, et al 1977) DISCOVER (1976), CHOICES and GIS (1978) as logical means to assist students in learning the process for making career choices. In addition, computer information systems provide occupational data, educational information and in some financial aid data. Total acceptance of computer systems in our culture has not occurred and many professionals remain ambivalent toward their use. In sum, counselors and teachers tend to be awed by their reputed power and somewhat uneasy with the implications. As Anastasio and Morgan (1972) note, professionals in education and human service agencies have resisted the encroachment of computers in what are seen as uniquely human function.

One of the major needs perceived by a variety of Maine state agencies was the need to develop a statewide system of career information. A cross department unit, the Maine Occupational Information Coordinating Committee (MOICC), consisting of representatives from Manpower, Education and Human Services was charged by the executive order of the Governor (August, 1978) to promote and develop a statewide system for career information. The MOICC group elected to field test the Guidance Information (GIS) in six public secondary schools and eight human service agencies in the Fall of 1979.

The purpose of this presentation is to report the impact of the GIS system on the program and management of the human service agencies during the Phase I field testing in 1979. More specifically the impact of GIS on the clients as well as the staff will be addressed.

PROCEDURE

The Guidance Information System (GIS) (1978) is a sophisticated career information system consisting of six files: Occupational Information, Armed Services Occupational, Two Year College, Four Year College, Financial Aid and Graduate School.

GIS is a product of thirteen years of development from 1966-79. It is currently being marketed by Time Share Corporation, a subsidiary of Houghton Mifflin Company and is intended for use by secondary level students and adults.

Eight widely diverse human service agencies were selected for the field testing. Two were correctional agencies, one serving youth and the other serving adults. There were three CETA sites, and a Community Action Program. Two other sites were regional offices of the Job Service and the Bureau of Vocational Rehabilitation.

A post-test only design was utilized to assess the impact of the GIS system on the participating sites. The site coordinators were asked to administer a User's Log to participants the first time they used the system. The form elicited some basic demographic information such as files used, age and sex and had seven questions for the individuals to evaluate their experiences. They were also asked to administer a Post Questionnaire to the group four to six weeks after their exposure to GIS. A total of 75 completed User Logs and Post Questionnaires were returned and analyzed.

Also, each site was visited by a staff member from the Center for Career Education, College of Education, University of Maine at Orono.

RESULTS

Only half of the eight human service sites provided post questionnaire data. Two were correctional centers, a CETA site, and the Job Service. One other site and the CETA site provided information from the User's Log. Ninety-three percent of the users found that the purpose of using GIS was clear to them. Eighty-eight percent reported that the directions for using the GIS guide and filling out the summary sheet were clear to them. One percent of the users right after using the terminal stated they disliked using it. Only about ten percent of the users reported that they had problems using the terminal. Three-eights of the sample said GIS gave them all the information they needed while three-tenths felt they needed more and about a third checked they were not quite sure.

Note: The authors are affiliated with the University of Maine at Orono, Orono, ME.

3 Twenty-two individuals from three agency sites completed the agency user's post questionnaire and thirty youth from the juvenile correctional agency completed the public school users' post questionnaire.

The following are some of the major positive results participants indicated from having used GIS:
1. It helped them clarify the educational plans they needed to make (73.7%).
2. It stimulated them to talk to people who are in the occupations they are interested in (68.4%).
3. It taught them a lot about occupations (68.4%).
4. It helped them to know how to divide occupations into categories or groups (52.6%).
5. It helped them learn a lot about themselves (42.1%).
6. It helped them learn how to explore occupations (42.1%).
7. It stimulated them to do some more reading about occupations and/or educational opportunities (42.1%).

The before and after comparison of the Agency User's Vocational Plans is presented in Table 1. The before and after comparison of their educational plans is presented in Table 2. It appears that there were fewer participants with vague ideas about their future vocational plans after they used GIS - a drop from 52.6% to 26.3%. The trend was for users to identify broad areas and then to narrow down their choices.

There was the same trend toward narrowing and sharpening educational plans but there were still about half of the group with vague ideas about their educational plans.

There were five items dealing with the client's evaluation of GIS. Overall, their ratings indicated that the information was adequate and gave them what they wanted to know. They felt the system was useful. No one checked that it was not useful and about one-third felt that GIS helped them enough that they felt all clients should have the same opportunity they did. Slightly over a third reported that their experience was useful and that they learned some things which helped them understand themselves better and to make good vocational and educational choices. Respondents felt that using the computer for guidance help was fun (57.9%) and a great way to get information to help them choose an occupation. Only two users stated that they felt GIS was boring to use. One felt that the computer was too slow.

Twenty of the participants wrote written comments. One major theme was that GIS helped them to increase their career knowledge and career awareness. (Example: Learned about the qualifications of job, schooling and work conditions; Learned to talk to people in different occupations).

The second major theme was GIS helped them in their decision making. (Example: Gave me a more realistic view of my plans for a career. Helped me to see what really was reality in going toward my goals. Helped me plan a route toward some specific career instead of leaving options wide open).

DISCUSSION

The two agencies which utilized GIS more systematically were the two correctional agencies. The counselors worked the use of the terminal into a career awareness course. They had a captive audience and were easily able to collect pre and post-test data.

The CETA and Job Service sites have a more transient population. It is harder for job service personnel to introduce users to GIS on a one shot basis and to integrate the system into existing procedures. Follow-up is difficult with the sometimes highly mobile population involved with the CETA sites. Vocational rehabilitation counselors didn't perceive that the system was appropriate for exceptional clients. Although inservice sessions were provided to the counselors at the sites, these sessions were primarily on how to access the system and exposure to the files and procedures to be used to access the system. In some of the human service agencies, there was the problem of staff stability. Some who were trained did not stay on their job and others were reassigned. Inclusion in the network was based on administrative decision rather than needs perceived by the counseling staff. They were expected to use it.

There was a delay in installing the system. It was installed several months after the inservice sessions took place and there was limited time to try out the system before the six months pilot period was up.

For the users the primary impact of GIS was on their career awareness. The majority of the users accessed the occupational files. These files tended to increase their knowledge of career fields. The computer appeared to be an interesting and motivational factor in the project. Very few clients had difficulties although they might have had apprehension about using the terminal. The system had some impact on goal setting and aiding in decision making skills in both the educational and vocational domain.

The majority of users enjoyed using GIS and found it interesting and valuable. The impact that GIS makes, however, is a function of the adjunct counseling or teaching program established by each site.

Recommendations
1. The inservice sessions for counselors on a computerized system must address the nature and needs of the clients within the agency.
2. A development program should be established for counselors to focus on career development elements such as career awareness, decision making, job seeking skills, job keeping skills, etc.
3. The computerized system is not going to be accepted by all counselors and all clients. There needs to be alternate strategies developed for counselors and users who prefer not to use the system. They should not be forced to use a computerized system if they don't want to do so.

4. The effectiveness of GIS or other systems relates to the type of guidance program at the site. There needs to be preparation for use as well as follow-up elements in the program. Some clients may be confused when they use the system. Provision needs to be made so that their clients have access to the counselor immediately to talk over problem areas.
5. There needs to be a continuous program of evaluation established at each site to assess the system, program, and its impact on users.

Summary

In essence, GIS received a favorable response by all users at the eight sites, but managerial problems proved to be a limiting factor. Potential adopters of computerized information systems will need to address the impact on staff functioning prior to installing a computer system. Philosophical values in relation to the use of computer technology proved to be subtle deterrent in changing staff roles and assignments.

Note: The presentation will include a panel discussion, selected handout materials and a demonstration of GIS. A two hour time block would be appropriate. Data from the Fall, 1979 Phase II testing will also be available for the April presentation, including Tables 1 and 2.

REFERENCES

Anastasio, E. and Morgan, J. Factors inhibiting the use of computers in instruction. Princeton, N.J.: EDUCOM: Interuniversity Communications Council, Inc., 1972.

Chapman, W., Katz, M.R., Norris, L., and Pears, L. SIGI: Field test and evaluation of a computer-based system of interactive guidance and information (2 Vols.) Princeton, N.J.: Educational Testing Service, 1977.

CHOICES: Employment counselor's manual. Employment and Immigration Canada, Occupational and Career Analysis Development Branch, 1977.

DISCOVER Foundations, Inc. A computer-based career development and Counselor support system, Westminister: Western Maryland College, 1976.

Guidance information system. Hartford, Conn.; Time Share, Inc. , 1978.

THE COMPUTER LABORATORY

John F. Schrage
John A. Sharp
Management Systems and Sciences Department
Southern Illinois University
Edwardsville, IL 62026

ABSTRACT

Originally, if a computer existed in an education institution, it was an administratively controlled center, but today the time-sharing and distributed processing concepts are reaching education. At Southern Illinois University at Edwardsville, there are four open computer laboratories besides the main administrative computing center. The largest facility, being the School of Business Laboratory, includes twenty-four terminals, of which twenty are cathode ray tubes (CRT); eight card punch machines; a verifier; and two dual purpose computers. The computers function as stand-alone computers and RJE stations into the two large computers on campus. Maintenance, supporting materials, and people have been the main problems, but presently that is generally under control. Although the Business Laboratory is open approximately eighty hours per week, the facility is just beginning to become used in the manner expected during its inception. Other laboratories are found in the Schools of Social Sciences, Education, and Science and Technology, with each having terminals into a Control Data Corporation CYBER computer.

INTRODUCTION

The first computer facility on campus followed the traditional educational computing centers equiped with International Business Machines (IBM) equipment. The center progressed from unit record through the IBM 1620, 1130, 1401, and 360 systems to the present IBM 370 Model 145 machine. The academic area needs expanded from two courses in 1966 to over fifty courses presently, and the use of the computer for administrative applications began to vie for computer time during the periods of classes. With other Illinois state institutions encountering that same situation, an educational network grant was approved for a group called the Illinois Educational Consortium.

COMPUTER COOPERATIVE

The grant created in 1974 a computing network of five state universities which has increased to the following nine state schools:

a. Chicago State University
b. Eastern Illinois University
c. Governors State University
d. Illinois State University
e. Northeastern Illinois University
f. Northern Illinois University
g. Sangamon State University
h. Western Illinois University

besides the Edwardsville campus of Southern Illinois University. The Carbondale campus dropped out of the cooperative in 1979.

The network was originally titled, Mid-Illinois Computer Cooperative (MICC), which has been changed to the Educational Computing Network (ECN). The purpose of the network organization is to provide computer services in support of the instructional and research programs at the member universities and other educational institutions throughout the state.

Hardware

The central computer facility of ECN is located in the administrative computing center on the Southern Illinois University at Edwardsville campus. The central processing unit was originally a Control Data Corporation CYBER 72-14 computer under the KRONOS operating system. During the 1979 summer months, the systems was upgraded to a CDC CYBER 170-730 using the NOS operating system. The specifications of the configuration now are:

a. 131,000 decimal words of main memory with sixty bits per word
b. ten peripheral processing units (PPUs)
c. two nine-track 800/1600 BPI model 669 magnetic tape units
d. CDC 501 line printer, 1000 lines/minute
e. CDC 405 card reader, 1200 cards/minute
f. four CDC 844-41 dual density drives, 238 million characters of storage/drive
g. two CDC 844-21 disk drives, 111 million characters of storage/drive
h. 1.174 billion characters of total online disk storage.

The NOS 1.3 level 485 operating system with a 64 character set provides four modes of job processing concurrently: time-sharing, remote batch, local batch, and deferred batch.

Data is stored internally as six-bit display code characters and ones complement binary numbers.

In addition, the system contains three data set controllers which allow simultaneous communication with up to sixteen remote batch terminals and 128 key-driven, interactive terminals. Two of the data set controllers provide asynchronous service to the member schools. A Gandalf PACX III system front-ends two of the data set controllers to expand the terminal access capability by allowing up to 254 terminals to contend for the 128 interactive computer ports.

Software

The software available from ECN to the users include the following languages:

a. APL
b. BASIC
c. COBOL
d. GPSS
e. LISP
f. PASCAL
g. PL/I
h. SNOBOL
i. SPSS
j. SIMSCRIPT
k. COMPASS (CDC assembler)
l. FORTRAN (CDC ANSI compiler and MNF instructional compiler).

Other special routines include the following:

a. CYBER MATH/SCIENCE LIBRARY
    FORTRAN-callable routines
b. CYBER Record Manager
    utility which standardizes input/output between different CDC product sets
c. CYBER SORT/MERGE
    utility which sorts and merges files
d. FOSOL
    matrix manipulation and statistical language
e. GIS
    guidance information system from Time Share Corporation
f. HELP
    ECN online documentation system
g. MPOS
    multi-purpose optimization system for linear programming
h. MULTITUTOR
    computer-assisted instruction language
i. PLOT10
    Tektronix FORTRAN-callable terminal plotting package
j. RIQS
    general purpose data base management system
k. RNF
    text formatter from the University of Illinois
l. SIMULA
    ALGOL-based simulation language
m. STAT
    basic statistical package
n. TEXT EDITORS
    EDIT, XEDIT, etc.

In addition, programs are available in the following areas from CONDUIT, a national organization for instructional materials:

a. Biology
b. Chemistry
c. Economics
d. Education
e. Management
f. Mathematics
g. Physics
h. Political Science
i. Psychology
j. Sociology

besides programs written by the instructors for use in the class.

Interface between Network and Campuses

To facilitate a smooth flow of correspondence between the central offices and member universities, a user coordinator, located on each campus, should be the first source of reference for any user who required assistance in the use of the ECN system. In order to guarantee that the system remains effective for its users, procedures must be followed to gain access to the system. The potential user is assign a user number and password before using the ECN facilities. The code is used to control usage patterns, allow the development of utilization statistics, and provide a level of security for all users. On member campuses, user number and password assignments are handled by the local user coordinator. Detail procedures for securing user numbers vary from campus to campus and prospective users must check with the local schools for specific information. The use is not restricted to the member schools previously listed. A variety of users also purchase time on the system for educational purposes.

THE LABORATORY

During the mid-1970s, the Edwardsville campus of Southern Illinois University was completing two new classroom and faculty office buildings for mainly the Schools of Education, Nursing, and Business. From the planning, the concept of the computer laboratory became a reality in early 1978.

The Business Computer Laboratory was to serve the faculty and students in that discipline and also provide a classroom atmosphere for teaching computer programming. A schematic of the physical arrangements (Figure 1) shows two large rooms for equipment, two carpeted rooms for card punch equipment; two medium-sized rooms for conference and library, and eight small rooms to serve as group project meeting rooms and/or offices.

Hardware

Based on a hardware and software study for the facility, the equipment proposed and purchased included the following:

a. eight IBM 029 card punch machines
b. twenty Hazeltine Modular One CRT terminals
c. four DECwriter II hardcopy terminals
d. CDC 1820 mini-computer (128K) with card reader and printer
e. WANG 2200 mini-computer (24K) with two CRTs card reader, and printer

The CDC 1820 computer had a dual purpose: (1) remote job entry (RJE) into the CDC CYBER system from ECN and (2) stand-alone small computer with FORTRAN and RPGII. The WANG system had a BASIC compiler and software packages in the financial area for class use. Each of the systems had the capability for possible RJE into the campus IBM system. Since the original laboratory equipment acquisition, an IBM 059 verifier was added to the card punch rooms. The link between the CDC 1820 and IBM 370 is now in operation in a batch mode. The general overview of equipment relationships can be seen in Figure 2.

Operations

The business facility was staffed with three to five graduate assistants from the School of Business and up to two student workers who provided tutorial assistance and supervision for all business areas including the computer courses. The responsibility for overall supervision of the laboratory was assigned to a faculty member in the Management Systems and Sciences Department on a rotating basis. Occasional technical support was supplied from the Academic Computing Office in the Computing Center. While usage of the laboratory was still limited, this arrangement was generally satisfactory.

As a service to the students, the business lab was open as follows:

8am to 11pm	Monday through Thursday
8am to 5pm	Friday
Noon to 5pm	Saturday
2pm to 7pm	Sunday

These hours complemented other hours of operations from the other laboratories such that the equipment was available at least eight hours per day.

The major users of the laboratory by the School of Business has been characterized in the following areas:

a. Computer programming in BASIC, COBOL, and RPGII

b. Statistical analysis using SPSS and other routines

c. Packages in the areas of forecasting, production, management, and marketing.

Problems

The use of the facility has also brought out problems. The original assistants and workers assigned to the laboratory learned by doing and became very valuable to anyone needing the lab and its services for research and instruction. In some cases, the assistants have been pirated from the laboratory. This still is somewhat of a problem, but diplomatically and slowly the situation is being corrected. The equipment purchased included only limited maintenance contracts thus repairs were done on a time and materials basis. Technical manuals and general supplies had the normal disappearance tendencies. With all of the problems, overall functions slowly became stabilized. The operation of all laboratory areas using computers is under study by a university-wide appointed Academic Computing Council.

During the 1979 calendar year, the laboratory situation was re-evaluated to provide students with a formal "open" laboratory environment.

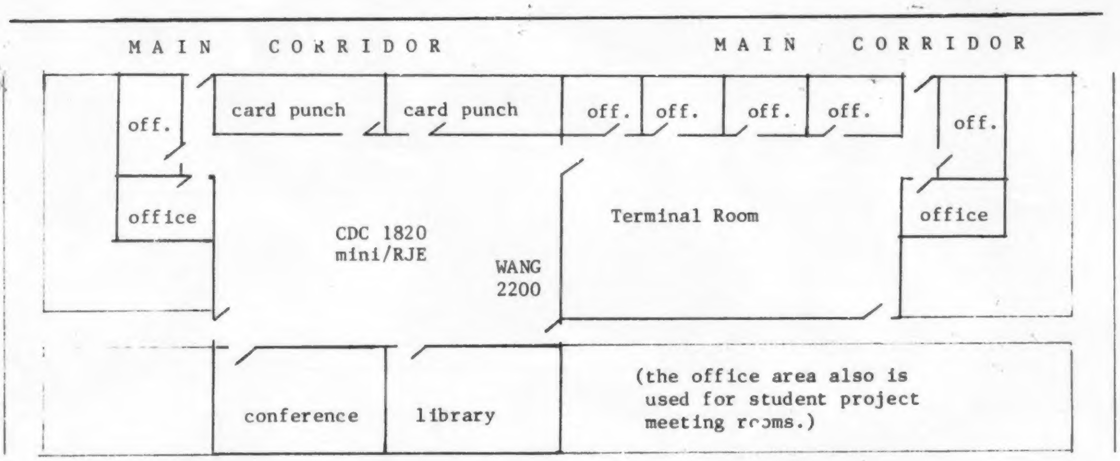
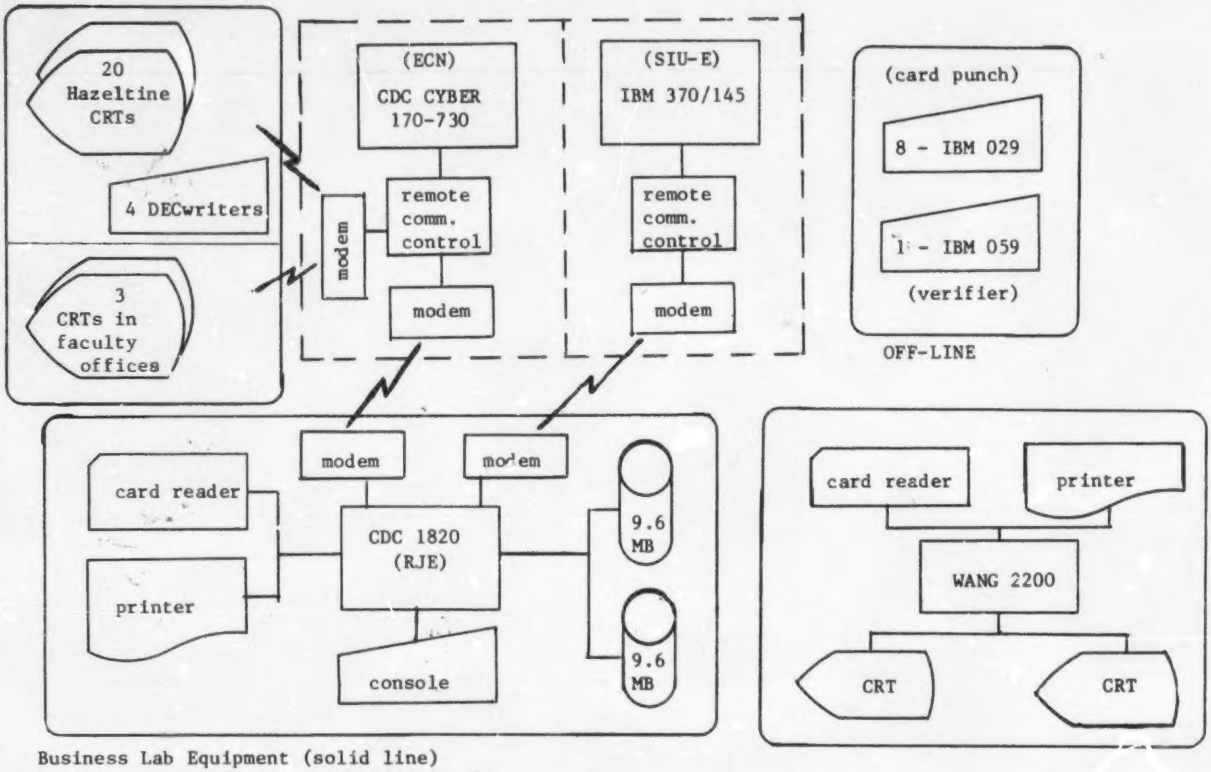


FIGURE 1

LABORATORY LAYOUT


20 Hazeltine CRTs
4 DECwriters
3 CRTs in faculty offices
modem
(ECN)
CDC CYBER 170-730
remote comm. control
modem
(SIU-E)
IBM 370/145
remote comm. control
modem
(card punch)
8 - IBM 029
1 - IBM 059
(verifier)
OFF-LINE
modem
modem
card reader
printer
CDC 1820 (RJE)
console
9.6 MB
9.6 MB
card reader
printer
WANG 2200
CRT
CRT

Business Lab Equipment (solid line)
Campus Computer Equipment (dotted line)

FIGURE 2

SCHOOL OF BUSINESS OPEN COMPUTER LABORATORY
Present On-Line or Time-sharing Concept

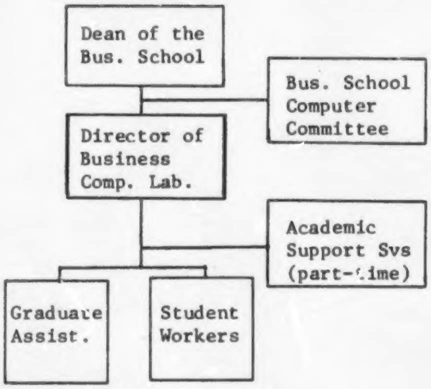
Previously there was an understanding that the facilities were only for the specified areas housing the equipment. Coupled with this was a procedure to show the distinctions between academic and administrative computing. Thus the academic equipment in the Computing Center was transferred to a Science and Technology laboratory. This gave the institution four academic laboratories with the Business School being the largest in all respects. The laboratories then existed in the Schools of Science and Technology, Education, Social Sciences, and Business.

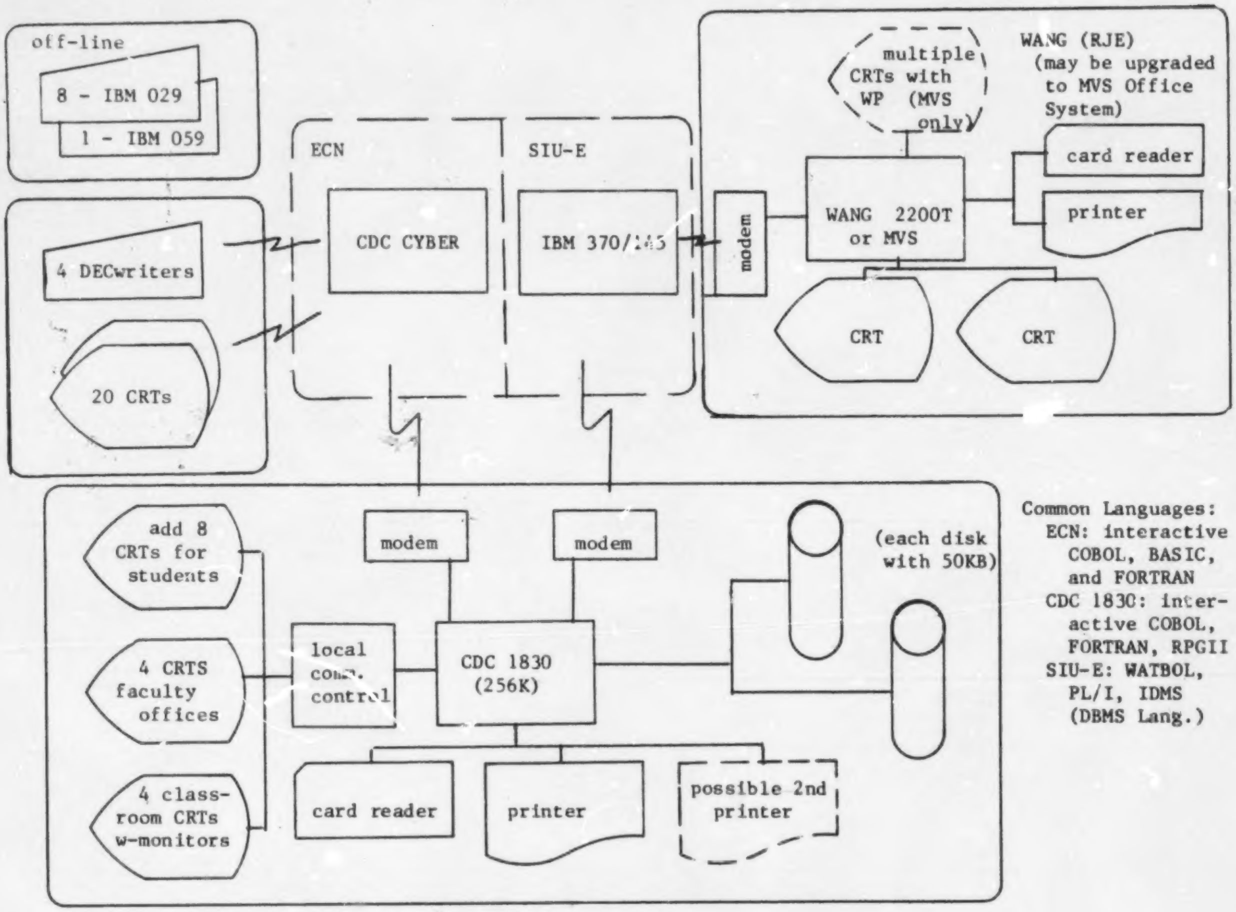
THE FUTURE

Sharply increasing demand for services and usage of the Business Computer Laboratory in its two years of existence has prompted a reassessment of the concept of the "open" laboratory and development of a new three year plan. The "open" concept has permitted a wide variety of academic uses of the computing capability with a minimum of direct administrative control. However, rapid growth has created a number of bottlenecks in the processing capability of existing computer equipment and the need for more direct supervision and additional technical services for academic support. The three year plan to meet the changes had subdivided needs and recommended solutions into three general categories. These are management and supervision, operating personnel, and computing capacity.

Management and Supervision

As competition for equipment and service has increased in the laboratory, the need for establishing priorities and policies for operations has increased. An organizational structure has been designed which will permit a rational and equitable distribution of services. This organization is:


Dean of the Bus. School
Bus. School Computer Committee
Director of Business Comp. Lab.
Academic Support Svs (part-time)
Graduate Assist.
Student Workers


off-line
8 - IBM 029
1 - IBM 059
4 DECwriters
20 CRTs
ECN
CDC CYBER
SIU-E
IBM 370/145
modem
multiple CRTs with WP (MVS only)
WANG (RJE) (may be upgraded to MVS Office System)
WANG 2200T or MVS
card reader
printer
CRT
CRT
add 8 CRTs for students
4 CRTS faculty offices
4 class-room CRTs w-monitors
local comm. control
modem
modem
CDC 1830 (256K)
(each disk with 50KB)
card reader
printer
possible 2nd printer

Common Languages:
ECN: interactive COBOL, BASIC, and FORTRAN
CDC 1830: interactive COBOL, FORTRAN, RPGII
SIU-E: WATBOL, PL/I, IDMS (DBMS Lang.)

Business Laboratory Equipment (solid line)
Campus Computer Equipment (dotted line)

FIGURE 3

SCHOOL OF BUSINESS OPEN COMPUTER LABORATORY
Three-Year Plan - Distributive Processing Concept

The Business School Computer Committee will be made up of representatives from various departments within the School of Business with proportionate voting powers according to usage. A member of the Management Information Systems faculty (a specialized area within the Management Systems and Sciences Department) and the Campus Director of Academic Computing will be appointed as technical consultants to the Business Computer Laboratory Committee. The purpose of this committee is to initiate and review long-term planning, establish broad policy and procedures, and oversee the general operations of the laboratory. Actual implementation of plans, policies and procedures, and supervision of day-to-day operations will be the responsibility of the Director of the Business Computer Laboratory, subject to periodic review by the Committee.

The responsibilities of the Director of the Business Computer Laboratory are:

a. to supervise and train graduate assistants and student workers for tutorial and operations duties in the facility.

b. to operate and supervise operation of the computer equipment in the facility.

c. to implement policies, write and maintain procedure files, and write and maintain documentation for many standard operations in the laboratory.

d. to provide technical support for hardware and software, and other functions, as required.

e. to monitor established priorities; to control usage; and to provide security for user programs, other software, equipment, and other laboratory resources.

f. to create a general atmosphere of professionalism and a level of operation of the facility commensurate with the best industry standards.

Tutorial and Operating Personnel

At the present time, tutorial and operations functions in the laboratory are carried on by graduate assistants and student workers. Whenever possible, these positions are filled by Management Information Systems or Computer Science students. This provides the student with actual work experience and, at the same time, provides valuable assistance to the many users of the laboratory. Nevertheless, there is an ongoing need for additional training of graduate assistants and student workers, especially as operating procedures become more complex under a distributed processing concept as planned in Figure 3. This need will be satisfied by training seminars conducted by the Director of the laboratory and interested faculty members.

Computing Capacity

The existing concept of time-sharing to a central host computer -- primarily the Educational Computing Network (ECN) -- is illustrated in Figure 2. In accord with industry trends and increased usage of the laboratory, the concept of time-sharing is being supplemented with the distributed processing concept (Figure 3). At the present time, the potentially powerful CDC 18xx is being used primarily as a remote job entry (RJE) to the larger CYBER system. As such, the system is hardly more than a non-intelligent terminal. However, two steps are being planned to better utilize the system which, without the changes, the system capabilities cannot be put into use. Step one is to upgrade this system in memory capacity, disk storage, and operating system to permit concurrent RJE processing and stand-alone capabilities such as RPGII, FORTRAN, and BASIC. A second step will be the addition of communication capabilities and terminals in oder to operate interactively on a local basis. These steps will rpovide a maximum flexibility in processing capabilities and backup in the event of a shut-down of the ECN CYBER system. At the same time, the number of terminals in faculty offices will be expanded and terminal monitors will be placed in four nearby classrooms for instructional purposes.

Thus, through the concept of distributive processing, users will gain substantial computing power while dependence on a central facility will be lessened. Many avenues of flexibility in processing and backup of operations, not now existent, will be opened up. While implementation of the distributve processing concept is in process (beginning this year), a feasibility analysis of the possibilities of integrating some more recent concepts of processing will be undertaken. Specifically, combined word processing - data processing systems, such as the WANG MVS Office Processing Systems, and various microprocessors will be studied.

REFERENCES

1. MICC User Guide and various updates to the overall system.

2. DIGIT (various issues), newsletter prepared by Educational Computing Network, Cottonwood Office Center, Cottonwood Road, Edwardsville, IL 62025. Walter Parrill, Director.

3. Sharp, John A. various correspondence on the Business Computer Laboratory, 1979-80.

Dr. Schrage is Assistant Professor of Management Systems and Sciences plus Coordinator of the MIS Faculty and acting Director of the M.S. in Management Systems. He previously served as coordinator of the Business Computer Laboratory. His work experience has run the gamit of the computer field from operations to management. He has published in the computer curriculum area at all levels with ACM, AEDS, IEEE, NBEA, and Society of Data Educators.

Dr. Sharp is Associate Professor of Management Systems and Science and Coordinator of the Business Computer Laboratory. His work experiences include several years as senior management consultant to industry and various positions of project and management responsibility with a major computer manufacturer. He has published two books and numerous articles in economics, management information systems, and interdisciplinary studies.

The address of the above individuals is:

Southern Illinois University at Edwardsville
Management Systems and Sciences Department
Box 106, Building II
Edwardsville, IL 62026

Phone: (618)-692-2504

THE DEVELOPMENT OF A MULTI-CULTURAL, COMPUTER ASSISTED READING PROGRAM

Richard A. Sheflin
Teaneck Public Schools
Teaneck, New Jersey

Introduction

The Teaneck, New Jersey, Public School System is located in northern New Jersey approximately five miles from the George Washington Bridge and New York City. Teaneck is a reasonably affluent district, with a student population of about six thousand pupils. The Teaneck community is racially, ethnically, and religiously diverse, with a non-white, minority school population of about forty percent. The Teaneck Schools provide a broad range of programs and services, and enjoy a good reputation for educational innovation and student accomplishment.

Academic computing is a well established service of the Teaneck Schools. Extensive programs in computer literacy, computer programming, and computer assisted instruction (C.A.I.) are supported at all levels of the school program. A Hewlett Packard 2000F computing system, along with other computer equipment, is installed at the central Instructional Computer Center, located at the senior high school. Remote terminals, primarily CRT's, are installed at elementary, junior high, senior high, and alternative school sites. The Teaneck Schools have received both national and local recognition for the excellence of their academic computing services.[1]

In 1977, a program to deliver significant C.A.I. services at the secondary level in the areas of remedial reading and mathematics was begun. This program was partially funded as part of a grant obtained under the Emergency School Aid Act (E.S.A.A.), as part of project G.A.I.T. (Getting Achievement and Integration Together). The achievement of minority students had been consistently below that of the white majority. The cognitive component of Project G.A.I.T. was designed to assist the integration of black and white students by narrowing the gap in achievement between the races.

Reading and math labs at the junior high schools were each equipped with threeCRT terminals tied to the central computing facility. Each reading lab was also equipped with a printer daisy chained to the terminals at that location. Reading and math tutors were hired at each location to supervise the C.A.I. facilities and to provide remedial and developmental instruction in coordination with the computerized program and the classroom teacher.

Computer terminals were also installed at the senior high school Learning Center for reading and math C.A.I. Tutors were hired to work in the Learning Center in conjunction with the terminals. New Jersey State Compensatory Education (S.C.E.) funds partially supported hiring the tutors.

The C.A.I. system became operational in October of 1977. Summer "Drop In" programs were offered to provide year round use of the C.A.I. facilities. The C.A.I. program has became a major component of the remedial reading and math effort at the secondary level.

After the 1977 school year, attention began to focus on the appropriateness of the commercially leased reading software. Although student, teacher, and administrative reaction to the C.A.I. program was consistently positive, suggestions for improved software were frequently being made.[2]

Many suggestions were obtained during informal discussions of the existing program with various staff members. In the course of conducting in-service workshops for reading and math teachers, other significant suggestions emerged. In addition, questionaires circulated to district staff members reflected the need to improve the present software.

After investigating commercially and publically available software in the area of C.A.I., it was the conclusion of the author that no significantly improved software was presently available from any outside source. Certainly, no available software package incorporated all or even most of the suggestions which had been obtained from the professional staff. It became obvious that the only acceptable means of acquiring the desired reading software would be to finance a local development effort.

Preliminary Planning and Funding

In December of 1978, preliminary planning for a local software development project began. A time line was constructed, providing a guideline for predicting the various stages of project development. This time line, along with estimates of required staff, were used to produce a projected budget.

It was decided that all software would be written to operate in the existing hardware environment. No money, therefore, was budgeted for equipment. The budget provided funds for employing the development team and for incidental supplies.[3]

The process of acquiring funding for the project, as dictated by the preliminary budget, was begun. Final approval for funding the development project as part of Project G.A.I.T. was obtained in February, 1979.

Staff Planning

During the three month period from January to March, 1979, a series of locally funded, in-service meetings were held with all secondary reading teachers, E.S.A.A. and S.C.E. tutors, and other appropriate teachers and administrators. The meetings were designed to ascertain the requirements of the professional staff for a new reading C.A.I. program, and to formalize the goals of the development project. The discussions focused on criticisms of the present programs, and on previously expressed concerns and suggestions of the staff.

The goal of developing a multi-cultural program required serious discussion at this formative stage. What, in fact, did multi-cultural mean? Surely, it did not simply mean the use of ethnically or racially identifiable names. The inclusion of the sentence "Juan stopped at the supermarket on his way home from school." as part of the reading text might identify Juan as a probable member of a specific ethnic group. It would, however, do nothing to increase cultural awareness or understanding among students.

Multi-cultural materials were defined by the staff as reading selections that related, in some non-superficial way, to a specific race, religion, ethnic group, or sex. Four specific types of passages were identified as providing an opportunity for the inclusion of significant multi-cultural materials in the reading package.[4]

1. Selections describing famous or interesting persons of a particular cultural group.
2. Selections describing historical events directly related to a specific cultural group.
3. Selections describing or examining social issues related to a cultural group.
4. Selections depicting the life-styles or cultural traditions of a specific cultural group.

There was agreement that a major early project of the author group should be to research the cultural and historical backgrounds of as many identifiable cultural groups as possible.

Many of the original suggestions for improving the existing reading package concerned the lack of management potential. The tutors and teachers associated with the C.A.I. program were becoming increasingly confident of their ability to utilize the computer in a sophisticated fashion. They desired more and better information from the computer, and more ability to direct and manage its use. The staff participating in the early in-service planning sessions articulated several important design decisions.

1. All student and staff services should be available on-line, directly to the pupils and staff at each remote location. Each CRT should be "daisy-chained" to an onsite printer, where reports, worksheets, class lists, etc. could be immediately available on demand. Teachers should enroll their own students and direct the use of all computer services themselves.

   This desire was in keeping with the already existing policy and goals of our C.A.I. efforts. All data management is done on-line, and all output is directed to the site from which it has been requested. In-service training of the staff has produced a generally high level of competance among the teachers and tutors.
2. The program should deliver comprehensive drill and practice in reading skills. The responsibility for primary instruction and remedial and developmental tutoring should rest with the professional staff.
3. The program should provide linear progress through a predetermined series of topics. Students should move rapidly through those skills where a high degree of accomplishment was demonstrated.
4. The computer should provide individualized student assignments based on the student's performance in each topic. Each pupil should receive, in printed form, selections and questions which were incorrectly completed, along with other materials of similar difficulty.

   These customized worksheets should be immediately available to to a student at the completion of his or her computer session. Additionally, worksheets should be obtainable for an entire group or class on demand.

These worksheets could be used in many ways, depending on the requirements and desires of the teachers or tutors. They could, for example, be assigned as classwork or homework. The teacher could also review the errors of a particular student for diagnostic purposes.

5. Teachers should have access to a wide range of statistical and diagnostic reports. These reports should be available on demand for individuals and for groups or classes of pupils.
6. Teachers should have the option of prescribing specific remedial or developmental sequences for particular pupils or groups of students.
7. The course should include work in the following topics: main idea; details; inference; vocabulary; study skills.
8. The computer should perform an item analysis of the questions in the course to assist with the identification of invalid or incorrectly placed materials.
9. The program should be usable for both remedial and developmental purposes.
10. The potential should exist for providing on-line standardized and diagnostic testing.
11. The program should be maintainable by regular staff members after the conclusion of the software writing project.

Final Project Definition and Planning

During February and March, 1979, final planning for the development project was completed by the author, in conjunction with the Director of Secondary Education and the District English Supervisor.

The size and composition of the staff were decided at this time. The project was led by the author, who served as Project Director, and also as the Chief Analyst/Programmer. Two additional programmers assisted with the software development.

Twelve reading teachers were chosen to research and author text for the various topics. Additionally, the English Supervisor was hired to serve as the proofreader and editor for the authored text. A typist/data entry clerk position was also included as part of the project staff.

An operative time line for development and testing was established. The schedule called for Version I of the software package to be available for limited testing by November of 1979.[5] Version I, in fact, came on-line in December.

The staff requirements being formalized during the concurrent staff planning sessions were utilized to design the basic software structure for the project. All programs were written in BASIC, the resident interpreted language of the Hewlett Packard 2000 system presently in use. The new software was designed to be compatible with the Hewlett Packard Instructional Management facility (I.M.F.) presently in use, thereby manimizing both developent time and the need for in-service training.

The programs and data base were designed to be information management oriented, providing the potential for significant statistical examination and comprehensive reporting. All programs were written in a modular form, facilitating the insertion of new routines and the alteration of completed software.

Software Development

The entire project staff met for the first time in April of 1979. The design decisions reached during the course of the past three months were reviewed, the time line and projected completion dates were discussed, and the division of labor among the members of the author group began.

Of the five topics previously identified for consideration, four were chosen for development at this time. Main Idea, Details, Vocabulary, and Study Skills were included, while Inference was left for later development. Vocabulary was divided into five sub-topics: synonyms; antonyms; homonyms; confusing words; and contractions.

A major discussion point involved the issue of "scope versus depth." It appeared obvious that the wider the range of grade levels selected for inclusion in the curriculum, the less comprehensively each grade level could be developed. The discussion centered around two dichotomous problems.

1. If the range of grade levels was too limited, an insufficient number of students would be able to utilize the course.
2. If the depth of content at each grade level was too limited, slowly moving students might not receive sufficient drill and practice at each grade level.

The participants decided that by including grades four through eight, and by delaying the development of the Inference topic to a later date, sufficient depth could be provided for the vast majority of students. The software was designed to insure that additional grade levels, additional materials at any grade level, or new topics, could be added at a later date.

Discussion also focused on the question of how precisely grade levels could be determined for the authored materials. With multiple authors, working on multiple topics, it was deemed advisable to be somewhat tentative in establishing grade levels for the authored text.

The staff concluded that the final

correlation between particular materials in the curriculum and precise grade levels should be delayed until the system was operational. At that point, item analysis and comparison to standardized test scores could assist in formalizing the grade level indicators.

Work began, on a part time basis, during the months of April through June of 1979, and continued, on a full time basis, during the month of July. Proof reading, data entry, and editing began in July, and continued into the fall of 1979. Additionally, a smaller authoring staff continued work during the 1979 to 1980 school year, on a part time basis. Development of the Inference topic, along with improvements and additions to the existing materials, continued through the fall and winter months.

The programming staff worked, on a part time basis, during April and May of 1979, developing the text editing and utility programs for the project. This software was tested during the month of June, and was put into operational use in July, when data entry and editing began.

Version I of the course driver, administrative, and report programs were written during July of 1979. Version I was tested and refined during the period from October through November of 1979.

Operational Testing - Version I

On December 1, 1979, Version I of the Teaneck Multi-cultural Computer Assisted Reading Program came on-line for operational testing. Tutors and teachers were instructed to enroll students in the course. They were cautioned, however, to avoid enrolling pupils who might become particularly upset or discouraged by any "bugs" or unusual procedures.

Version I contained materials from three of the four developed topics. The Main Idea and Details topics were entirely complete. The Vocabulary topic contained only materials from the homonyms and contractions sub-topics. Study skills were not included.

Each topic underwent a rigorous proof reading sequence. All original long hand materials were proof read and then entered into the system. All text was then printed, proof read, and edited twice before it was deemed acceptable for operational use. The expert proof reader altered the content of the curriculum where necessary, as well as correcting grammatical and spelling errors. Materials were moved, replaced, or deleted wherever appropriate.

Editing of the curriculum materials continued on an on-going basis during the operational testing period.

Four sub-versions of the Version I software were implemented during the two month operational testing period. Versions 1.0, 1.1, 1.2, and 1.3 each contained minor software corrections, minor alterations suggested by teachers or tutors, and improved formats for the presentation of the course materials.

Version I incorporated the following significant features.

1. A reactive and intelligent driver controlled student progress in each topic, based on his or her previous performance.
2. Statistical information summarizing a student's performance during a session, as well as cummulative results, were provided as hard copy at the conclusion of each student session.
3. Individualized student worksheets were immediately printed at the end of each student session.
4. A limited administrative program permitted the enrollment of students at the beginning of the course or at any point within each individual topic.

The staff responded positively to the Version I software. Teachers specifically praised the improved statistical reports and the clear, concise nature of the student worksheets. Teachers also confirmed that a high degree of student interest existed for the new software. The new materials were perceived to be challenging to a wider range of students.

The tutors and teachers using the program expressed the desire to have access to the next version as quickly as possible. They were anxious to utilize the improved management facilities of Version II.

Operational Testing - Version II

Version II, which became operational in February of 1980, incorporated major additions to the error recording component of the software, providing automatically scheduled review of previously missed questions and reading selections. Students prepared for the periodic review lessons with the help of the individualized worksheets provided by the computer.

Antonyms and Confusing Words were added to the Vocabulary topic in Version II. The new software incorporated many improvements reflecting staff suggestions related to Version I. Each student worksheet, for example, included the date on which it was produced.

Operational Testing - Version III

Version III, scheduled for operational testing during April and May of 1980, represents the final software product of the present development project.

A sophisticated administrative program highlights the improvements in this final version. Teachers can optionally assign concentrated practice from particular topics. This increases the usefulness of the program as a developmental tool for students with diagnosed reading problems.

The new administrative component also provides for variable session length, variable acceptable response times for different students, and other control options.

Synonyms are included, thereby completing the Vocabulary topic. The two final topics, Study Skills and Inference, are also available as part of the Version III curriculum.

Project Completion

In June of 1980, the development project will be effectively complete. The final version of the software will be thoroughly tested and ready for regular use. Teachers' manuals, indexed curriculum guides, and documentation for the utility programs will be complete and available for staff use.

In September of 1980, the software developed as part of this project will replace all commercially obtained reading and language arts software presently in use in the Teaneck Public Schools. The financial savings to the school district will be substantial, and the software will more accurately reflect the ethnic, racial, and religious diversity of the Teaneck community.

Although the project itself will be over, the ability to maintain and improve the system will remain a responsibility of the Teaneck Schools. The editing and utility programs, developed as part of the project, will provide an efficient means for updating, correcting, and adding to the curriculum materials.

It is anticipated that reading and English teachers will, as part of regular in-service work, continue to improve the reading materials. Several present staff members have been part of the project since its inception. They should provide the continuity and expertise necessary to follow through on improvements to the curriculum.

The software produced by this development project should continue to play a major role in the remedial and developmental reading instruction of the Teaneck Public Schools for years to come.

Conclusion

There are several aspects of Teaneck's efforts in this project that deserve special note and emphasis.

1. The project was born in response to a specific local need. The Teaneck Schools, after successfully experimenting with C.A.I., were looking for a more appropriate software package, incorporating the suggestions of the local professional staff.
2. The expertise and hardware required to produce a sophisticated instructional package were already in place.
3. The project was completed using only local personnel as part of the development team.
4. The final product is individually tailored to the special needs of the Teaneck, New Jersey, community.
5. The final product will be maintainable on-site. It should enjoy a long operational life.
6. A local development project was necessary because little has been done at the state, regional, or national level to support the development of economically affordable C.A.I. services.

The marriage of the computer to the national priority of improving reading and language arts skills is an idea that deserves more attention and support. Computer Assisted and Computer Managed instructional services should be available, at reasonable cost, to all school districts.

It is the hope of this author that this chronical will provide some impetus to the development of new computer hardware and software, designed to provide effective and attractive computerized instructional services.

Endnotes

1. Teaneck High School, for example, was named as an "exemplar" in a 1977 study of academic computing conducted by the Human Resources Research Organization, and funded by the National Science Foundation.

	VERY EFFECTIVE	EFFECTIVE	SOMEWHAT EFFECTIVE	INEFFECTIVE
Effectiveness of present system	8	17	8	2
Potential effectiveness of system	28	7	1	0

Responses of professional staff members familiar with the C.A.I. system

Table 1

2. The professional staff (teachers, tutors, and administrators) consistently felt that the potential for the C.A.I. system was outstanding. When staff members familiar with the C.A.I. system were asked to rate the effectiveness of the present system, and also the potential effectiveness of a highly developed C.A.I. system, their responses were overwhelmingly positive. (See Table 1.)
3. The development project was originally funded only through June of 1979. It was hoped that some form of income could be obtained to continue work past this date. Funding as part of the E.S.A.A. program was eventually obtained through June of 1980.
4. The following passage, for example, meets the criteria to be classified as multi-cultural.

   "Beckwourth Pass through the Sierra Nevada Mountains was named after James Beckwourth, a black explorer and frontiersman, who helped lead pioneers of the middle 1800's into the West. He became famous as a scout, trader, and suprisingly, as an Indian chief; for he lived with the Crow Indians for almost six years and became one of their most respected leaders."
5. The goal was to utilize the new software in an operational environment during the 1979 to 1980 school year. This would provide several months of on-line evaluation and testing.

FIVE STEPS FOR ACQUIRING YOUR NEXT COMPUTER

N. Michael Slater, CDP
Director, Data Processing
Kanawha County Schools

INTRODUCTION

One of the most important tasks for which a data processing manager is responsible is the selection of computer hardware. Adding peripheral devices or off-line equipment is not unusual, but selecting an entirely new system is a task not often undertaken. This paper is a discussion of a process for selecting a new computer system. It is certainly not a unique procedure but rather a personally defined and organized approach which incorporates basic techniques familiar to all managers.

The computer selection process involves five basic steps:

1. Identification and Definition of Needs
2. Request Proposals
3. Proposal Evaluation and Hardware Selection
4. Hardware Recommendation
5. Installation Planning

This process is independent of computer size and can be applied equally well to the selection of a very small computer or a very large one. It is effective in selecting a first-time system or in replacing an existing one. While the process is designed to be used in a competing vendor situation, all five steps are applicable in dealing with only one vendor.

STEP ONE - IDENTIFICATION AND DEFINITION OF NEEDS

It is the data processing manager who must decide that, for whatever reason, new hardware is needed to carry out the responsibilities of the department. Hardware is in reality a tool to assist personnel in accomplishing assigned tasks. It is desirable then to identify the specific tasks for which the hardware may or will be used within a projected time period of at least three but not more than six years. This period will roughly correspond to the life expectancy of the hardware.

Since data processing departments are service agencies, it is logical that users should play a major role in determining specific tasks to be accomplished by the data processing department during the projected time period. A list of tasks (applications) including current tasks which will be continued, plus new tasks deemed desirable by the user, must be developed. After securing approval from the top administrator, schedule appointments with all major department heads. This is in effect a sales technique; it is important that you go to the department heads rather than having them come to you. Explain the purpose of your meeting and ask the department head to confirm the desirability of continuing existing data processing services and identify possible new services or modifications to existing applications. Be helpful and offer suggestions of possible new applications you believe could be beneficial to the respective department. Accept all realistic ideas suggested by the department head regardless of personal feelings.

After each interview, write a formal memorandum to the department head itemizing all existing and potential applications identified as needs. Don't forget requests for major revisions to existing applications.

This memo serves three purposes:

1. It provides a ready review reference.
2. It documents what was discussed and requested.
3. It builds goodwill for current and future working relationships.

Upon completion of the interviews select applications for inclusion in a formal "Summary of Needs Survey" list. For proposed applications, this list also becomes a priority ranking with each application being assigned an estimated date of development within the proposed time period previously defined. Selection and ranking should be based on potential benefit to the school system and obvious need. Do not eliminate any reasonable item, but give low priority to items of marginal value.

The Summary of Needs Survey is the cornerstone for building a case for new or additional hardware. It may be used to solicit support from top administrative levels and most importantly, to provide evidence of planning.

STEP TWO - REQUEST FOR PROPOSALS

You are now ready to present to all potential vendors an invitation to submit a hardware proposal. In addition to an appropriate cover letter identifying when and to whom the written proposal should be submitted, the invitation should contain a copy of the Summary of Needs Survey, a description of currently installed hardware, a brief outline of current and proposed data files, and a written outline of the minimum items required to be included in the vendor's written proposal. This last document is typically referred to as a Request for Proposal or an RFP.

The purpose of the RFP is to request information necessary to evaluate availability of technical support and costs which are directly or indirectly related to the proposed hardware. It should not dictate the type hardware to be proposed. That choice should remain with the vendors since they best know their product lines. The RFP will serve as a guide for determining if all proposal requirements have been met and is the principle component for evaluating and comparing proposals equitably.

The RFP should contain specifications for: (1) Hardware, (2) Technical/Software Support, and (3) Conversion. Each major section should be divided into subsections and each subsection should stipulate information required from the vendor concerning availability, support, and costs related to that subsection.

The section dealing with hardware should include subsections concerned with equipment, support service, usage costs (lease basis) and facilities requirements.

The following example illustrates the type of statements to which the vendor should respond.

> *Hardware Equipment*
>
> *Please quote:*
>
> A. *Detailed description of main processing unit and each peripheral component*
>
> B. *Delivery availability*
>
> C. *Growth potential in terms of future upgrade capability*
>
> D. *Hardware price quotation - both lease and purchase*
>
> E. *Approximate freight charges related to hardware delivery*
>
> *Hardware Support Service*
>
> *Please quote:*
>
> A. *The availability of trained local service personnel*
>
> B. *The maximum amount of time required to receive replacement parts from remote inventory locations*
>
> C. *The hours of the day free service is available (if applicable)*
>
> D. *The availability of after-hours service and at what cost*
>
> E. *Preventive maintenance time required in terms of hours, frequency, and what hours of the day preventive maintenance would be scheduled.*

Statements should be simple and direct in order to avoid communication problems while allowing the vendor the freedom to be innovative.

Technical/software support is a more difficult section to prepare since there is greater diversification among vendors, but this increases its importance. This section should include subsections concerning programming assistance, application packages, programming software, operating system software and online data communications. As an example:

> *Programming Assistance*
>
> *Please state the availability of vendor programmer/analysts for technical assistance including the location of such personnel and the related costs for their services.*
>
> *Operating System Software*
>
> *Please identify:*
>
> A. *A description of the proposed operating software*
>
> B. *A statement concerning support and updating policies*
>
> C. *A statement of costs as related to the above two items*
>
> D. *The minimum and maximum memory requirements for the operating system including primary memory as well as secondary storage requirements.*

Again the statements are direct and simple, related primarily to availability of support and costs. Some items may not necessarily fit this criteria. Item 'D' under Operating System Software is an example. This item would also be important primarily in terms of performance. Any item which may be directly related to your particular applications should be included in the appropriate section/ subsection.

The conversion section should contain subsections on personnel training, file conversion, software conversion (if applicable), and hardware conversion (if applicable). The following example is given because of its significance in evaluating costs due to the wide variance of responses you may receive from vendors.

Hardware Conversion

Please identify:

A. *The estimated time for hardware paralleling and costs for new hardware during parallel conversion time*

B. *The location of proposed hardware during parallel operation.*

A fourth section simply titled "Other" should also be included in the RFP. In this section the vendor should be invited to identify any non-requested items related to hardware, technical/ software support or conversion which the vendor would like to have considered in the evaluation of the proposal.

STEP THREE - PROPOSAL EVALUATION AND HARDWARE SELECTON

After all written vendor proposals have been received, you are ready to begin evaluation. A word of caution may well be appropriate at this point. Care should be taken at all times not to inadvertently discuss one vendor's proposal with another vendor. No consideration should be given to a vendor whose proposal is submitted after the stated due date in the letter of invitation. You cannot give the appearance of favortism or advantage to one vendor over another.

Vendors normally like to make a formal oral presentation when submitting their written proposals. This action is not necessary but is acceptable as long as all parties understand that it is only the written proposal which will be evaluated.

After the formality of accepting each proposal, begin the evaluation by working with one proposal at a time. Read the proposal through; then prepare a written understanding (yours) of the proposal or a Vendor Response Summary, using the RFP as a guide. State your understanding of what the vendor has said in response to each item in each section/subsection of the RFP. This should first be done in draft form and then reviewed with the vendor. Clarify and modify the draft copy to the mutual satisfaction of both parties if possible. After the revisions are incorporated, prepare a final copy and submit it to the vendor for acceptance. Upon acceptance, terminate discussion of the RFP with that vendor and begin the review of the next vendor's proposal.

While preparing the Vendor Response Summary it is important to work with each vendor as if that particular proposal will ultimately be accepted. In so doing you are more likely to be attentive to detail and better understand what is really being proposed.

Upon completion of the Vendor Response Summary, a cost analysis should be prepared. Using the RFP once again as a guide, identify all direct and indirect costs associated with a given vendor's proposal. Summarize the costs by line item of your existing data processing operating budget. Next compare each line item to your existing budgeted line items and determine net increases and decreases for those line items effected. Costs should then be extended over a three-year period to evalute long term as well as immediate effect. Upon completion you should have a cost evaluation analysis for each vendor's proposal as is illustrated in Figure #1.

At this point it is now known what effect each proposal will have on your budget if implemented, or, stated in a more popular concept, "what's the bottom line".

Do not overlook any indirect cost reduction that would result from implementing a given proposal. It is important that all cost considerations be taken into account.

There is one final analysis to be made prior to hardware selection. This analysis is also based on the original RFP document. Compare each proposal by item within each section/subsection of the RFP and rank the proposal 1, 2, 3, in terms of desirability as illustrated in Figure #2.

This evaluation should cause one proposal to surface as the most desirable.

Now decide on one proposal which will best do the job defined in the Summary of Needs Survey at an acceptable cost. Select only one. The choice will be well documented and can be defended. Expect challenges and be prepared to defend your choice with the documentation you've assembled.

STEP FOUR - HARDWARE RECOMMENDATION

At last! Many weeks if not months have been spent talking, evaluating, writing, thinking, asking questions and answering them. The decision has been reached but the job is not done. The task now is to convince the adminstrators and ultimately the governing board that the decision is sound, wise, and in the best interest of the school district.

This can best be done by using the simple approach. A one-page memorandum of recommendation should be written as concisely as possible, reviewing the procedure that has been used and naming the vendor's proposal that has been selected. The attachments included with the memorandum are the supporting evidence The attachments are taken from the documentation which has been created from the beginning of the quest. These should be:

(1) Summary of Needs Survey
(2) original RFP (Request for Proposal)
(3) copy of each Vendor Response Summary
(4) By Item Analysis
(5) Cost Evaluation Analysis.

The purpose of the one-page memorandum is to allow the reader to quickly receive the message without doing a lot of reading. The attachments provide evidence of the planning that has gone into the effort, the thoroughness of the evaluations, and an audit trail as to how you arrived at your recommendation. Most top level administrators and board members will not care to read the attachments but it is important to provide the detail so that the reader has an opportunity to independently decide if all vendors have been treated equally and fairly.

Depending upon the administrative formalities of the school district, a formal presentation to the board may be required. If a formal presentation is requested or required, it should be brief. While selection of hardware is an important and significant event in the life of a data processing manager, it is not necessarily so to a board member. The board most likely has already decided informally to accept and support the recommendation prior to official action. It is not necessary to sell your selection.

STEP FIVE - INSTALLATION PLANNING

As data processing manager, the responsibility for the successful installation of the new hardware is yours. This also means that if things don't go well you will be given the majority of the blame regardless of how much help was received in making the hardware selection. This is really as it should be for it is during the installation of the new equipment, especially with a major hardware conversion, that the data processing manager's ability is truly tested.

Planning is the real key to installing successfully and on time. Form a planning team of representatives from the vendor's staff and your own. Members of this team should be key personnel who will be directly responsible on a day-to-day basis before, during, and after the installation. This team should begin meeting as soon as possible after the hardware selection has been made and meet frequently, preferably once a week.

The first task of the planning team is to prepare a formal Schedule of Events. This schedule should be detailed and include not only the obvious but also the seemingly insignificant things that are critical to the actual operation of the computer once it arrives. Each event should be given a target date for completion. The schedule should then be arranged in chronological order for easy reference.

The Vendor Response Summary makes a handy guide to identify those things that should be included in the Schedule of Events. It will also help in settling questions as to what the vendor is committed to do and who is obligated to pay for what. Avoid dealing with the vendor, however. as an adversary. At this point cooperation on both sides is needed to achieve the common goal of a successful installation.

Assign responsibility for every event to the appropriate member or members of the team in a formal manner so that there is individual responsibility for each event. During the teams' regular meetings require members to give progress reports indicating any problems that may be encountered in meeting an upcoming target date. If an event appears to be in danger, determine how it can be brought up to schedule and act immediately. If it becomes impossible for an event to be completed by its target date, reschedule it, but don't forget to reschedule all future dependent events if necessary.

As the team coordinator, follow up on all potential problems. See to it that things are done on time.

CONCLUSION - PITFALLS AND ADVANTAGES

Acquisition of a computer is somewhat like life - there are pitfalls along the way. It would be impossible to foresee every problem, but there are three that experience dictates mentioning.

First, there is the temptation to say, "This procedure looks good, but I do not have the time and/or the experience to do it. I'll hire a consultant." It is true you can hire a consultant who is more knowledgeable (and a lot who are far less), but when the consultant leaves so does the knowledge. Hardware selection is an education in itself. The in-house staff must ultimately shoulder the responsibility. If the in-house staff does the selection, there will be a greater understanding of the hardware to be worked with and a much greater commitment to make it succeed. As for time - take time. Nothing affects the job of a manager more quickly than hardware performance or lack of it. It is too important to the manager to let someone else make the selection.

The second problem is lack of adequate staff. Obtaining a new computer without adequate staff, both in terms of quality and quantity, has probably been the downfall of more computer installations than all other reasons combined. Good staff can compensate for poor hardware, but the best hardware in the world cannot compensate for poor staff. Hire the best personnel available, giving preference to quality over quantity.

The last pitfall is the temptation to save time, effort or money in hardware selection by taking shortcuts. All five of the steps outlined for hardware acquisition must be dealt with either formally or informally. A formal documented procedure forces the necessary planning and avoids problems caused by errors of omission and miscommunication.

Now for the good part - the advantages. An approach such as this provides for a totally planned procedure. It avoids the "where do we go from here" syndrome. In the early days it was common practice to get a computer and then determine what could be done with it. That is no longer acceptable.

A second advantage is one of the simplest but most important. The Cost Evaluation Analysis provides knowledge of costs before commitment. This is a cardinal rule of good management yet sometimes forgotten where computer acquisition is involved.

A third advantage is that prior to commitment there is a mutual understanding between the customer and the vendor as to the obligation and responsibilities of both parties. This leads to a good working relationship which pays further dividends to all concerned.

Finally, the use of these five steps will make you a better data processing manager. How? As a manager you will have greater control of "what's happening" which is what managing is all about.

FIGURE #1

"Vendor A" Cost Evaluation Analysis

Line Item	Cost Increase FY81	Cost Increase FY82	Cost Increase FY83

FIGURE #2

Comparative Analysis - By RFP Item

Hardware Equipment	Vendor A	Vendor B	Vendor C
A. Hardware Configuration			
(1) Processor	3	1	2
(2) Memory Size	2.5	1	2.5
(3) Memory Utilization	2	1	3
(4) Card Reader	3	2	1
(5) Line Printer	3	2	1
(6) Disc Drive - Speed	3	1	2
(7) Disc Drive - Storage	3	1	2
(8) Magnetic Tapes - (Equal)	-	-	-
(9) CRT Unit (no judgment)	-	-	-
B. Modular Growth Potential	3	1	2
C. Price	1	3	2
Hardware Equipment - Overall	3	1	2

COMPUTER POLICY IN HIGHER EDUCATION
MAJOR STATEWIDE ISSUES

Edward P. St. John
Associate Dir. of Research and Planning

Missouri Department of Higher Education

COMPUTER POLICY IN HIGHER EDUCATION MAJOR STATEWIDE ISSUES

At their August 1978 meeting, the Missouri Coordinating Board for Higher Education approved the formation of a statewide Computer Policy Task Force. The charges given to the task force were: to create an atmosphere of resource sharing; to improve the cost effectiveness of computing services; and to identify areas of unmet needs and consider new and/or innovative ways of using computing to meet those needs. While the Computer Policy Task Force (CPTF) conducted a comprehensive study and developed a statewide plan that addressed these concerns, it also identified several other major statewide issues regarding the future of higher education computing in this state.

The Computer Policy Task Force study was the first comprehensive effort to develop a statewide plan for higher education computing in Missouri. By 1975, twenty-six states had initiated or completed statewide planning activities. At that time, Missouri was listed as one of ten states where there was no planning or no immediate prospects (Heydinger and Norris, 1977, p. 35). Most of these early plans addressed the issue of expanding computing capability; most led to the formation of computer hardware networks. Since many of these early plans were completed, computing has changed, especially in the availability of low-cost computer hardware; particularly mini and micro processors and very large main frames with extensive memory and storage capacities. In fact, by 1976-77, computer hardware was no longer the single largest portion of computer expenditures for higher education (Emery, 1977). Instead, personnel had become the largest expenditure, representing 41% of the total expenditures, contrasted to 22% for hardware leases and 18% for hardware purchases (Hamblen and Baird, 1978). In this new context, statewide planning for higher education computing has to address more than hardware planning issues.

Statewide planning for computing in higher education is entering a new era as a result of these shifts. While the initial planning emphasized hardware planning, a new generation of statewide planning efforts will, out of necessity, emphasize a broader range of policy issues.

Major Statewide Issues

As a result of the Missouri study, the Computer Policy Task Force identified several issues that should be of on-going concern to state-wide, institutional and, perhaps, federal policymakers in higher education. This paper addresses some of these issues, first as general concepts, followed by information generated in the Missouri study which is used to elaborate on the implications of these issues for Missouri. (For a complete discussion of the results of the Missouri study, refer to the Report to the Coordinating Board for Higher Education of the Computer Policy Task Force, August, 1979: Jefferson City, Missouri. Hereafter CPTF, 1979). The policy issues considered here are: technological developments, pedagogical issues, management systems developments, financing policy, statewide planning, and organizational considerations.

Technological Developments: Two dimensions to the development of computer technology are critical to higher education policy: first is the actual development of new technologies, and second is their adaptation to higher education. Currently, the rate of technological change in the computer industry is rapid, resulting in the increasing availability of low-cost mini and micro computers. The real breakthroughs in low-cost hardware are currently in the small powerful machines, which are not always readily adapted to education, due to a number of factors influencing the change process. In higher education in general, the adaptation of the new technologies has been selective; research and administrative applications have been adapted more readily than have the instructional potentials. If the potentials of computing for improving the quality of instruction and the efficiency of administration is to be realized, then more coherent statewide, and probably federal, policy will be necessary.

In Missouri, there is an uneven picture. Overall, increases in the level of expenditures for higher education computing have not kept pace with national trends during the 1970's. The University of Missouri system has made the most extensive use of computing and has an elaborate computer network for its four campuses. The regional universities, state colleges, and metropolitan two-year college systems have developed stand-alone computer facilities that meet many of their local computing needs. Since each of these latter schools are not part of a statewide system, they have not developed hardware networks that extend beyond the campus or the district, unless they have joined the University of Missouri network. The level of service provided by these stand-alone facilities does not equal the level of service, in either administrative or academic areas, provided by the University of Missouri network. Potentially, these installations can exploit the University of Missouri's specialized capabilities. Several of the other institutions in the state, especially Harris-Stowe State College and the non-urban community colleges, have not yet been able to exploit the potentials of computing for academic or administrative uses; many currently have no access to computing resources.

As a result of its study of higher education computing, the Computer Policy Task Force identified three major issues confronting continued development of these technologies. First, several public sector institutions have not yet gained access to computing resources, and providing such should be an objective for state policymakers. Second, the potentials of instructional computing were under-utilized by all sectors of public higher education; as a result, planning for a statewide (or regional) instructional time-sharing system is an alternative that needs to be considered. Third, while administrative uses of computing were more pervasive than any other use, there are many potentials for cooperative systems development in this area as well.

Pedagogical Issues: The potential pedagogical implications of computers provide the most compelling reason for utilizing this technology in education. We have reached a situation in our society where school children can learn through the use of computers at home, while their school experiences emphasize the traditional student-teacher mode of instruction. Teachers are usually not equipped to deal with the needs of such students. Certainly, the implications of computer technology for formal education--in elementary, secondary, and higher education institutions--have not been fully realized. However, without coherent national and state policies in these areas, it is likely that other organizations, perhaps the computer industry itself, will become the primary learning centers. If the new technologies are to become a more formal part of education, then training of teachers now employed by schools and colleges is a critical issue. Researchers have found that a variety of internal organizational factors often limit the adaptation of instructional uses of computing in higher education institutions (Rockart and Scott Morton, 1975).

The majority of the state-supported higher education institutions in Missouri make some use of academic computing. An overall differentiated pattern is evident, with the greatest use at the UM campuses, followed by the other four-year institutions, and the least use by faculty at the two-year institutions. In response to task force surveys, all three sets of institutions identified financial, hardware, and personnel constraints as factors limiting the academic uses of computing. Training of existing personnel is one area of need identified by most institutions, especially in the two-year and four-year institutions. Two types of training needs were identified: in-service training for computer center personnel and training for faculty in instructional computing techniques and applications used in their disciplines.

In one survey, all presidents gave uniformly high ratings to objective statements that emphasized the use of computing in instruction. Surveys of academic administrators and of computer center directors, however, found that the quality of instructional computing facilities and the levels of use of instructional applications were relatively low, with a few notable exceptions, particularly at Northwest Missouri State University and at the University of Missouri-Rolla. The Computer Policy Task Force recommendations emphasized the need for a comprehensive statewide study of academic computing needs. The purpose of the proposed study would be to identify areas where cooperative development of instructional computing capabilities could take place. One possible area for statewide actions is the development seminars that familarize faculty with applications that are commonly used in their disciplines. Such support services will be necessary, in addition to providing access to hardware, if instructional uses of computing are to become a reality.

Management Systems Development: Administrative computing has been more readily adapted in higher education than instructional applications. In fact, a reverse situation often exists; colleges and universities often try to implement systems that are more complex than their management needs justify. There have been three waves in the development of management information systems (MIS) in higher education, from data processing in student and financial systems, to integrated systems for enrollment and financial forecasting, to the eventual development of complex financial and student flow modeling systems. One problem that many institutions encounter when they try to develop the more sophisticated applications is that basic systems are not yet in place and, therefore, the capacity to support these more complex applications does not yet exist (St. John, 1977; St. John and Weathersby, 1980). In fact, researchers are now pointing out that management information system planning should be based on careful study of institutional management needs (Coleman and Topping, 1979; St. John, 1979). Statewide policy should take into account institutional management needs when planning for management information system development.

There was considerable evidence that continued development of administrative systems was needed by institutions in each segment of state-supported institutions. A summary of the status of MIS systems for each segment shows:

> University of Missouri: All of the University of Missouri campuses have already developed management information systems. They have some system in production in each area. A variety of operating modes are used: batch, on-line, combined batch on-line. The MIS development priorities include new systems and refinement or modification of systems that are already in production. The UMR, UMKC, and UMSL campuses have moderate to well-developed MIS--with some automated systems in most areas. The UMC campus has a well-developed MIS and is currently considering some refined systems such as induced-course-load-matrix analysis and student-flow modeling that were not being considered by any other institutions in the state.
>
> State Four-year Institutions: There was more variation in the state of development of the four-year institutions. Three of the regional universities--Central, Southeast, and Southwest--have one or more automated systems in production in most areas and they are considering modification or refinement of some of these systems. These institutions are likely to be using a mixture of processing modes: batch, on-line, and combined batch on-line.
>
> A second set of four-year institutions--Lincoln, Southern, Western, Northeast, and Northwest--have some of these systems in production in some of the areas identified on the questionnaire. These institutions identified new systems as their top MIS development priorities, with the exception of Southern, which wanted to convert existing systems from batch to on-line processing.
>
> The remaining four-year institution, Harris-Stowe, has no automated system and did not respond to the MIS survey.
>
> Two-year Institutions: The most extreme variation in the current status of MIS development is within the two-year sector. The two urban community college systems maintain automated information systems for the campuses and for central administration. For these campuses, production systems are operating in most areas. St. Louis Community College's central office identifies refinement or modification of existing systems as the top MIS development priority.
>
> Several of the other campuses have a few initial systems in production, usually using combined manual and computerized processing. Most of the non-urban two-year campuses make limited use of automated MIS. Usually they have contractual arrangements with outside firms or relatively modest in-house development if they have any automated systems. Several operate with entirely manual systems. Most consider computerization relevant for some systems (CPTF, 1979, pp. 59-60).

The Computer Policy Task Force recommended that a major study of institutional uses of administrative computing that specifically considered the possibilities and potentials of compatible information systems at state-supported institutions, of specialized services at some institutions, and resource sharing (or computer hook-up) capabilities. Certainly there is need for caution in this study in order to avoid overemphasis on high-cost systems; however, there are potentials for cost savings when shared development is undertaken. In any event, such a study, and subsequent policy considerations, must be sensitive to the different MIS needs of institutions in the state. The study results should help to inform future decisions to update administrative software systems and to avoid excess duplication of these systems.

Financing Policy: Educational financing in the United States is complex; it involves federal, state, local, and private decision-makers. The development of public policies that will encourage formal education institutions to adopt the new technologies will involve policymakers at each of these levels. A federal policy that advocates increased use of computers in higher education will have little impact without financial incentives for state- and local-government agencies, which fund most public education. Further, the lack of growth in education anticipated during the next decade, decreases the incentive for institutions to use the new technologies, especially if there is not a coherent federal policy to this area. In the absence of such a federal commitment, states will need to take a more active role in financial planning for efficiency in the use of computer technology.

The financing of higher education computing is complicated further by institutional financing practices, which often do not provide incentives for efficient use of computing resources. Most higher education computer centers are budgeted installations that do not bill for services (Hamblen and Baird, 1979). There is little incentive for these installations to provide low-cost services. Out of convenience, computer centers usually increase their service potential by going for the larger machines. The greatest potential for cost savings, however, is the area of fastest development in the computer industry, micro and mini processors (Emery, 1977). While there are a few examples of direct institutional policies aimed at taking advantage of these potential cost savings in higher education (Massy, 1979), this practice is the exception rather than the norm.

In Missouri, the institutions are subject to diverse funding regulations and constraints. The University of Missouri system has had a more favorable situation than other institutions in the state: as a multi-campus system, it has been able to take advantage of the economies of scale of large machines in network configuration; also, its computer hardware decisions have not been subject to review by the Office of Administration/Division of Electronic Data Processing Coordination (OA/EDPC). The state universities and colleges have individually developed their computing capacities under the review of the Office of Administration. Administration at these institutions view this review as an extra layer of bureaucracy. In spite of the review by OA/EDPC, the institutions have not yet taken advantage of the economies of scale that statewide networks would provide; nor have they taken advantage of the lower cost of small- and medium-sized machines now available. The two-year colleges have not been subject to OA/EDPC review. However, with the exception of the two urban systems, these institutions have lacked the critical mass and financial resources necessary to develop their computing potentials.

Nowhere in the state has there been a single focal point for planning, especially for planning that would maximize the economic benefits of current trends in computer and communication technologies. The Computer Policy Task Force recommendation emphasized the need for statewide financial planning. The task force did not recommend a change in the role of the Office of Administration in the approval of equipment requests. Certainly a statewide acquisitions policy needs to be developed that will take advantage of potential cost savings.

Statewide Planning: Computer planning will undoubtedly become an increasingly important issue in the education policy arena. A first phase of planning, which many states undertook in the early and middle 1970's, was for the development of computer hardware. While computer hardware strategies remain an important planning issue, especially given recent innovations in telecommunications and computer technologies, the new wave of state-level planning is emphasizing the problems of adaptation; i.e., training and supporting faculty and administrators in the use of automated instructional and administrative systems.

Missouri was one of a few states with no prior history of planning; therefore, both sets of planning issues were present--the need for hardware development at underserved institutions and the need for greater cooperation in the development of computing services. The task force, however, recognized that the actual hardware configurations were of secondary importance, since there are several possible ways of providing services to underserved institutions and of providing incentives for cost savings in hardware lease and purchase decisions at the institutions. The more pervasive need appeared to be providing services and training to academic and administrative users and incentives for using computers, especially in instruction.

The Computer Policy Task Force recommendations emphasized the need to establish a mechanism for statewide policy planning for computing. Many of the issues identified during the course of the task force study would need on-going attention in the policy arena.

Organizational Considerations: The utilization of computer applications in education often requires adaption of organization structures in higher education, and

211

probably in other levels of education as well. The advent of interinstitutional computer hardware networks is but one example. Increasingly, the addition of computer support networks is important as well, since every institution cannot afford to develop the full range of support services necessary for instructional and, perhaps, administrative computing. However, the development of support organizations at regional, statewide, or national levels requires modification of existing governance and management practices in education.

The principle assumption underlying the Computer Policy Task Force recommendations was that computer hardware is not the major obstacle to achieving statewide objectives. While there are some hardware deficiencies in the state, in fact some institutions have no computing facilities, these limitations are not as much of an obstacle to achievement of the statewide objectives as the coordination and personnel training problems facing all institutions in the state. The CPTF survey results indicated that faculty and staff development and training are one key to realizing the potentials of computing in both the academic and administrative areas. Another problem has been the limited computer-resource sharing to date among institutions in the state which has resulted from the absence of a statewide cooperative mechanism for planning and development. The task force concluded that these problems face institutions in each sector of Missouri higher education and that solutions can be developed on a cooperative basis.

In considering the functions which a facilitating organization might assume, it is important to understand that the whole is greater than the sum of its parts. A facilitating organization would be better able to provide assistance in purchasing and selecting equipment by virtue of its participation in statewide planning. Even if the institutions in a state perform one or more of the major functions of a facilitating organization in an effective and coordinated manner, the state can profit from a clearly identified, active facilitating organization. According to the CPTF Report:

> One of the most important activities of the facilitating organization would be training faculty and staff, especially those of the small institutions. In the instructional area, there are needs for training in computing techniques as applied to specific academic disciplines as well as the obvious requirements for training in programming and computer science. Resource persons from institutions already heavily involved in academic computing should be utilized in this effort. Thus, the central staff need not have expertise in a variety of disciplines since the objective can be accomplished by organizing workshops and arranging for qualified institutional personnel to make presentations. Equally important, however, is the availability of follow-up technical consultation and general personnel support; this would have to be provided or coordinated by the central organization staff. Without some new organization providing such services, it is not likely that the less developed institutions can be rapidly brought up to the present state-of-the-art in academic computing. Certainly the problem is not solved by simply placing computers and/or terminals at all of the institutions (CPTF, 1979, p. 71).

The essence of the Computer Policy Task Force recommendation that a statewide facilitating organization for higher education computing be created in Missouri, was that many of the problems identified during the study cut across traditional organizational boundaries. Further, the task force members agreed that there simply were not enough financial resources available to higher education institutions in the state to develop these independently. Since computer and telecommunication technologies do not lend themselves to traditional organizational boundaries, the task force decided that the support services necessary to adapt these technologies to education could be most effectively and efficiently developed on a cooperative basis. The purpose of the facilitating organization would not be to develop these services centrally, but to facilitate their development on a cooperative and shared basis at the institutions in the state.

The Computer Policy Task Force did not specifically identify the organizational placement of the facilitating organization. Several alternatives were considered: placement at one of the state institutions, in the Office of Administration, in the Department of Higher Education, and the formation of a not-for-profit corporation. While the CPTF's analysis of these alternatives speculated that placement in the Department of Higher Education might be a necessary first step (CPTF, 1979, pp. 73-74), it did not make a specific recommendation about its placement. Instead, these considerations were left to the Coordinating Board for Higher Education.

Conclusions

The Computer Policy Task Force's study of computing was an important first step. It identified several major statewide issues that confront computer policy in this state. The main thrust of its recommendations was to establish a new statewide organization that would achieve the following:

a. To serve as an information clearinghouse for education computing. This function may include maintaining listings of computer hardware and software and administrative and academic applications, publishing a newsletter, disseminating information from national computing organizations, and serving as a focal point for Missouri participation in national resource-sharing organizations.

b. To facilitate the development and support of academic and administrative computing. This function may include workshops that introduce faculty to computer applications in their disciplines, workshops that concentrate on special administrative applications, and training programs for data-processing personnel.

c. To provide a formal voice for higher education computing in the state. This function may include encouraging planning for higher education computing and assessing achievement of statewide objectives.

d. To encourage active development of appropriate levels of computing resources at all colleges and universities. This may include encouraging interinstitutional cooperation in the development of new computing capabilities at all institutions (CPTF, 1979, p. 76).

Due to certain exogenous factors, especially the time constraints of the state budget process in Missouri, initial implementation of the Computer Policy Task Force recommendations progressed rapidly and without full consideration of the policy issues identified by the task force. The plan was distributed in August 1979. At their September meeting, the Coordinating Board for Higher Education approved the inclusion of a budget item for fiscal year 1981 for implementing the studies recommended by the task force. At their October meeting, the Coordinating Board decided to appoint a Computer Facilitating Committee to serve as policy advisory group for higher education computing and to implement the studies proposed by the Computer Policy Task Force. This action, which was based on the recommendation of the CBHE Advisory Committee, departed from the CPTF recommendation in two ways. First, the name of the key computer facilitating group was changed from the Missouri Center for Higher Education Computing (MoCHEC) Policy Board to reflect the Advisory Committee's concern that this committee not become a statewide computer network. Second, the membership of the committee was changed to include college presidents rather than lower level administrators.

The Missouri Coordinating Board for Higher Education considered the policy issues raised in this paper at their December 1979 meeting; at their January 1980 meeting, the Board appointed an eight-member computer facilitating committee, to serve as the key policy group in this important area, and gave the committee the initial charge to develop a plan for addressing the statewide issues identified here. It is hoped that this committee will evolve into an organization that can deal effectively with statewide computer policy issues in Missouri.

REFERENCES

✓ Coleman, D. and Topping, S. "A Model for Assessing the Effectiveness of a State University System Management Information System, Presented: 1979 AIR Forum, San Diego, California.

Emery, J. C. "The Coming Challenge of Campus Computation," EDUCOM, Vol. 12, No. 4, 1977, pp. 12-18.

Hamblen, J. W. and Baird, T. B. "FICHE Bytes: An Executive Summary" EDUCOM, Winter, 1978, pp. 2-5

✓ Hamblen, J. W. and Baird, T. B. Fourth Inventory of Computers in U.S. Higher Education, 1976-77. Princeton, New Jersey: EDUCOM, 1979.

✓ Heydinger, R. B. and Norris, D. M. Cooperative Computing: A Process Perspective on Planning and Implementation. Princeton, New Jersey: EDUCOM, 1977.

✓ Massy, W. F. "Policy Issues in Networking," in J. L. Emery (Ed.) The Reality of National Computer Networking: Proceedings of the 1978 EDUCOM Fall Conference. Boulder, Colorado: Westview, 1979, pp. 59-72.

✓ Missouri Department of Higher Education, Report to the Coordinating Board for Higher Education of the Computer Policy Task Force. Jefferson City, Missouri: MDHE, 1979.

Rockart, J. F. and Scott Morton, M. S. *Computers and the Learning Process in Higher Education*. McGraw-Hill, 1975.

St. John, E. P. "MIS Development in Higher Education: A Framework for Systems Planning," Presented: 1979 AIR Forum, San Diego, California.

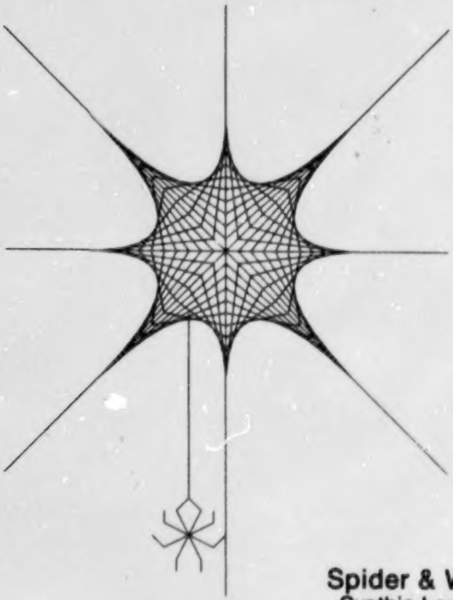
St. John, E. P. "Structure, Management, and Intervention: Final Report, "*A Study of Selected Developing Colleges and Universities*. Cambridge, Massachusetts: HGSE, 1977.

St. John, E. P. and Weathersby, G. B. "Management Development in Higher Education: Intervention Strategies For Developing Colleges and Universities," *International Journal on Institutional Management in Higher Education*. July, 1980, Vol. 4, No. 2.

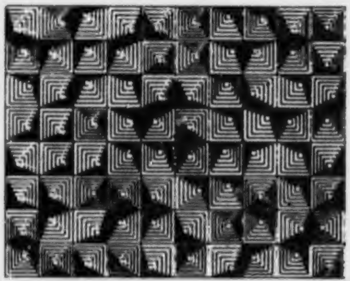
[illegible] Bill

Plots Created by CSC74 Students

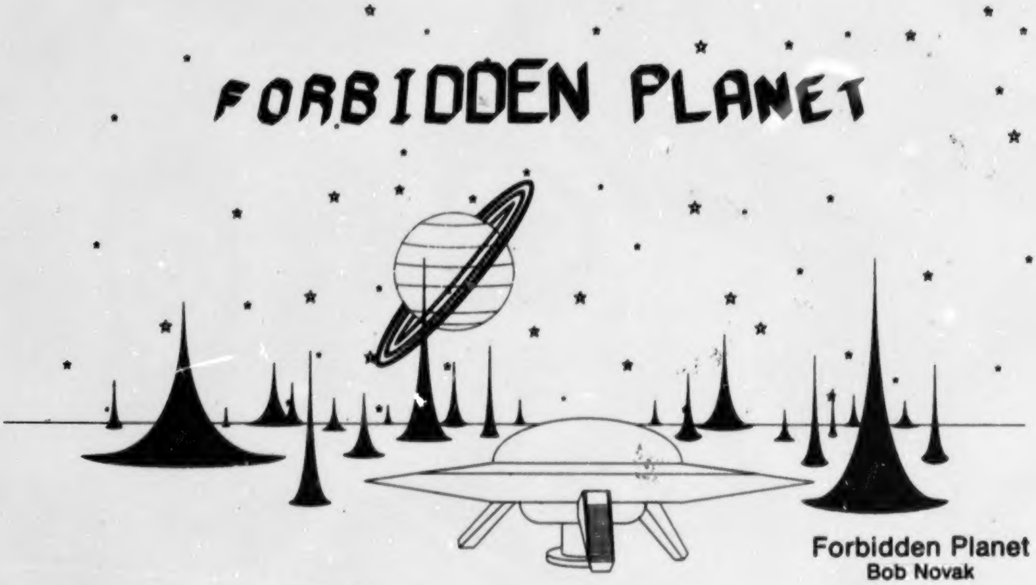
University of Missouri - Rolla



Spider & Web
Cynthia Louise



Random Squares
George Endress


FORBIDDEN PLANET

Forbidden Planet
Bob Novak

MICROCOMPUTERS--PROMOTING THEIR USE IN ELEMENTARY AND SECONDARY SCHOOLS

Patricia Sturdivant
Region IV Education Service Center
Houston, Texas

Computer assisted instruction (CAI) has already demonstrated that it can play a major role in improving the effectiveness of instruction. Because of its capability to provide interactive, individualized instruction, computer assisted instruction represents a powerful medium for any instructional domain. The emergence of microcomputers has removed cost as the primary obstacle to widespread use of computers for instruction. A recent report in the Washington Post indicated that nine out of ten U.S. homes are expected to have a personal computer by 1985.[1]

This report makes no attempt to review the various microcomputers available or the software packages that run on them. Rather, it represents an effort to survey the impact of microcomputers on education. It is a state-of-the-art paper that synthesizes current developments in the educational use of micros and presents some of the administrative considerations that should be taken into account.

Specifically, this report:

- summarizes growth trends in the micro market
- reviews the state-of-the-art as it applies to instruction
- explains how recent advancements in technology are affecting costs
- defines hardware and software considerations
- details current courseware development
- presents some of the ways micros are currently being used in public schools
- presents future trends

MICROCOMPUTER TECHNOLOGY

The use of microcomputers in education has been a rather recent development. Only in the past two years have microcomputers been marketed to the business and education communities. Despite the slow start, most experts predict that, in the next few years, use of microcomputers will increase dramatically. The impact of the microcomputer has been called the Second Industrial Revolution--man's brain power is being multiplied in the same manner that machines multiplied his muscle power 100 years ago.[2]

Recent curiosity about computers has prompted some confusion about terminology. Micro-, mini- and mainframe are terms that denote the relative size of the processing architecture of a computer. One definition is based on bit size. If the processor is on one chip and uses an 8-bit "word," it is a microcomputer. If it uses a 16-bit word, even if it is on a chip, it is classified as a minicomputer. A 32-bit or larger machine is a mainframe. With present day advances in electronics and data processing, the smallest microcomputers can easily perform most of the same instructions as the largest mainframes. The basic differences are speed of execution and the maximum amount of data that can be stored at one time.

Microcomputers are currently used in many different fields. The main advantages of microcomputers are affordability, transportability and dedication. Their cost is decreasing each year, while their capability is increasing. Because micros are independent of a mainframe system, many communications related problems are eliminated. With the ever increasing cost of phone lines, this factor is a significant one.

Dedicated use is an advantage because it contributes to cost effectiveness, especially for the small user. Large systems that are used for a variety of data processing tasks often limit the user's access to a mainframe system.

There is no doubt about the micro's potential for affecting educational change, but the technology is still so new that it is difficult to determine what kinds of developments will take place in the next few years. There are now about 40 computer companies producing microcomputers. These companies are emerging and faltering overnight. Some predict that within two or three years, most of these companies will go out of business. Only about five may survive. This factor makes the manufacturer's longevity an important consideration.

---

[1] Leon Nawrocki, "Personal Computing and Exportable Training", Society for Applied Learning Technology Conference Proceedings. (Warrenton, Virginia: Society for Applied Learning Technology, 1979). p. 89.

[2] Carlo Pastore, "Technical Training Systems Using Microcomputers", Conference on Microcomputers in Education and Training. Arlington, Virginia, May 17-18, 1979.

COST/TECHNOLOGY CONSIDERATIONS

Computer assisted instruction has already proven itself effective in impacting student achievement. Significant gains have been documented at the elementary and junior high levels.[3] Despite these positive results, many school districts still find the costs of computer assisted instruction prohibitive.

Computer Curriculum Corporation (CCC) has dominated the field for several years. It is well known that CCC has little competition. Most of the computer mainframe manufacturers who ventured into this market (e.g. Hewlett Packard, Olivetti, IBM) have lost money.

The only major company left is Control Data Corporation (CDC), which is plunging ahead in its effort to market PLATO, a structurally sophisticated system that is motivating to students. Unfortunately, CDC has not been successful in promoting PLATO usage in public schools. Only eight school districts in the entire nation are using PLATO because the cost is still astronomical. After six years of operation, the educational division of CDC is still running in the red.[4]

Low-cost microcomputers have already had a profound effect. In fact, they are turning the economics of computer assisted instruction around. Radio Shack, for example, has already made a computer, keyboard, cathode ray tube and cassette tape interface available for $600.[5]

According to a recent WICAT Report, computers will revolutionize education. Labor costs (teachers' salaries) are going up while technology costs are decreasing at a phenomenal rate. This trend will continue becuase semiconductor manufacturers are able to place more and more electronic components (transistors, resistors, capacitors and diodes) on a small one-fourth inch piece of silicon. Every year since 1960, the number of electronic components that can be placed on a single piece of silicon has doubled. The economic implication of this doubling process is that the price remains relatively constant after the technology of a new doubling has been mastered.[6] By the mid 1980's, powerful computers, with up to a million bits of memory, can be put on one chip costing less than $100. One can now purchase a version of a computer that would have cost more than $25,000 a decade ago for less than $1,000. This system would include a microprocessor, cathode ray tube, keyboard, cassette player, BASIC language and floppy disc capability.

Just a few years ago, the simplest electronic calculator sold for several hundred dollars. Today, a basic calculator can be bought for less than $10. It is obvious that the personal computer will follow the same pattern.

It seems hard to believe that the first microcomputer was made only four years ago in 1975. By 1982, it is projected that one micro chip will be able to remember and recall about a quarter million bits or basic units of computer information. The number of logic, or decision making, computer circuits on a chip will climb to roughly 25,000 by 1987.[6] Like the calculator, the cost of the microcomputer will come down dramatically while its capabilities steadily increase.

CURRENT MICROCOMPUTER DEVELOPMENTS IN EDUCATION

Several of the major micro companies are pursuing the educational market with vigor, as indicated by these recent developments:

- Apple Company, Inc. (Cuppertino, California) has developed a microcomputer that is widely used in education. APPLE II, noted for its color graphics, has a built-in keyboard that includes video display circuits, microprocessor, power supply and memory integrated circuits on a single printed circuit board. The company has demonstrated its interest in the education market by donating $250,000 in cash and computers to a newly formed nonprofit foundation that will promote the use of computers in education.
- Bell & Howell Company (Chicago, Illinois) is distributing a special classroom version of the APPLE II microcomputer. According to Richard Fowkes, president of the company's Audio Visual Products Division, "Education is clearly moving to more self-paced instruction. The microcomputer is the audio visual device of the 1980's." Bell & Howell has recently contracted with Merrill to develop micro-based computer assisted instruction curriculum.
- Tandy Corporation/Radio Shack (Fort Worth, Texas), which markets the TRS-80 microcomputer, has sent out packets of promotional literature to every school district in the country. The company launched a massive pre-Christmas campaign aimed at getting parents to buy a TRS-80 "tutor" for their child.
- The Dallas Independent School District, through the Foundation for Quality Education (Dallas, Texas), has developed a math curriculum (1-6) for the TRS-80 which has been promoted on a national level. The program sells for $995 per cassette pack. Tandy provided the Dallas Independent School District with a special accommodation in agreeing to add a protective device to its computer. In making this special modification, Tandy helped Dallas protect their proprietary rights to the software. The added feature prevents the user from getting a printout of the actual program set.

---

[3] "Computers Are Fun, But Can They Teach?" Educational Testing Service Report, Vol. XXV, No. 4, (Winter, 1979).

[4] "The Micros Take to the Classroom," Business Week, June 4, 1979, p. 104 D-F.

[5] Radio Shack 1979 Catalog, (Fort Worth: Tandy Corporation, 1978), p. 79.

[6] "Vast Computing Power is Seen as More Circuits Squeeze on a Tiny Part, Wall Street Journal, April 27, 1979.

- Minnesota Educational Computing Consortium (MECC) signed a contract with Apple Company, Inc. in January 1979. MECC has already purchased 900 micros for Minnesota schools. This institution leads the nation in making the transition from a mainframe to a micro delivery system for computer assisted instruction.[7]
- Control Data (Minneapolis, Minnesota) recently set up a microcomputer division. The company has contracted with micromakers to adapt its PLATO courseware to small machine cassettes. (Rental costs for a single PLATO terminal still run about $1,100 a month. At this price, one can buy an APPLE II microcomputer a month for what it costs to rent one PLATO terminal for a year.)
- Commodore Business Machines (Palo Alto, California) PET microcomputer has been promoted to school people through a special "3 for the price of 2" sale. With this incentive, many educators have bought PET's even though the memory capability is limited.
- Mattel Toys (Hawthorne, California) intends to use the home terminal as a springboard into the home computer market. The company plans to introduce (for about $250) a full-keyboard microcomputer unit that can be combined with the terminal to form a home computer. A language cartridge will be added next year that will allow the user to write his own programs. (Mattel has a contract with Texas Instruments for micro-chips.)
- Telesensory Systems, Inc. (Palo Alto, California) has developed some innovative computer applications for the handicapped. Telesensory's game center consists of eight micro games for the unsighted; oscillating tones replace the screen markings for contests like paddle ball and synthesized speech is used for games such as tic-tac-toe, blackjack and skeet shoot.
- Xerox Research Center (Palo Alto, California) is presently developing a notebook sized microcomputer. Its language (SMALLTALK - a new type of language dealing with parallel messages) will allow children and inexperienced adults to write their own computer programs.
- Utah State University (Logan, Utah)is piloting the use of a video disc microcomputer system in teaching mentally retarded children. The video disc--a shiny, silver-colored disc that looks much like a record--has feedback response capabilities coupled with the motivational power of television.
- WICAT (Orem, Utah) has recently undertaken development of a reading program which will be delivered via microprocessor. It is being written in BASIC language.
- Mountain Hardware has developed a product called "Supertalker"™ for the APPLE II computer. With it, human speech can be digitalized for storage into random access memory (RAM). Supertalker™ plays back digitalized, high quality speech through a loud speaker under program control and opens up tremendous possibilities for reading and language instruction.
- Whitney Educational Services (San Francisco, California) is developing a system which will connect an APPLE II computer running computer assisted instruction programs to a Sony Betamax video cassette recorder. Actual television pictures are used in addition to computer text and graphics in this interactive computer assisted instruction system.
- Learning Technology Institute, under a grant from the United States Office of Education's Bureau of Education for the Handicapped, has developed the Visual Image Controlled (VIS-I-CON) projection system which consists of a frame-addressable super eight cassette, a rear screen projector, a programmable read-only memory for program instruction and a microprocessor. The unit can function on a stand-alone basis to deliver instruction or it can serve as a computer terminal or a host machine that drives other equipment.
- Northwest Regional Laboratory (Oregon) will soon receive a grant from the National Institute of Education for the purpose of establishing a software clearinghouse for microcomputers.
- Prescription Learning (Chicago, Illinois) has developed software in both elementary reading and math. The programs already developed are primarily used with students who qualify for compensatory education.
- Textbook Publishers are beginning to explore the potential of producing learning materials based on microcomputer technology. Mattel Toy Company recently purchased the MacMillan Corporation, which is a significant development since Mattel is already using micro chips for their educational toys.[8]
- Milliken (St. Louis, Missouri) is the major company that has launched into full scale production of microcomputer software. Milliken, which is only a year old, has developed a math curriculum which operates on the APPLE, PET and TRS-80. Their operation is unique because they are converting their computer assisted instruction curriculum so that it will operate on the different equipment presently in the educational marketplace.
- Texas Instruments (Dallas, Texas) announced their new microcomputer (TI 99/4) in June 1979. TI has their competitors worried because of their developments in voice synthesis technology. The new microcomputer uses "GROM" cartridges, which prevent the user from duplicating the programs. Disc drives are to be available in February 1980. Last year TI introduced the "Speak and Spell," an educational toy that features a 200-word vocabulary stored on a

---

[7] Kevin Hausmann, "Statewide Educational Computer Systems: The Many Considerations", Creative Computing, V [illegible] September 1979) pp. 82-85.

[8] "Mattel to Acquire MacMillan, Inc." Houston Chronicle, August 31, 1979.

solid state memory chip. The device sells for just $75. The "Talking Language Translator," to be available for $250 in September, will offer the user access to pre-programmed phrases. These can be combined with words to form thousands of phrases. It can translate 1,000 words from input language to output language. Programmed drills will be partially beneficial in helping the language student in English, Spanish, French, German, Japanese and Chinese.

- Science Research Associates (SRA) is piloting a micro-based classroom management system in Kansas City, Missouri.
- Courseware, Inc. (San Diego, California) has developed a classroom management system for the APPLE II. Courseware is a software development company that is negotiating with the California State Education Department for a minimum competency curriculum which can be accessed on a microcomputer.

IMPACT ON EDUCATORS

Educators are becoming increasingly aware of the impact of microcomputers on education. However, the state-of-the-art in software lags far behind hardware. Despite hardware advancements, reliability is not what it should be.

Schools will not be able to resist microcomputers because most of the major vendors are promoting them through the home market. This inroad will inevitably have a dramatic impact on the schools.

PRESENT MICRO UTILIZATION IN PUBLIC SCHOOLS

Interest in microcomputers is steadily increasing. Educators who have had experience with timesharing systems are finding the microcomputer particularly appealing. Microcomputers are viewed as solving some of the common problems experienced with large scale computers: (1) high cost of computer technology, (2) complex installation requirements, (3) lack of local control over computing resources and (4) escalating telecommunication costs. The microcomputer, though it lacks the sophistication of the larger system, is perceived by many as a solution to the drawbacks of a larger timeshare system.

There is some evidence to suggest that schools are accepting the use of microcomputers before the resources are available to support their use. Many districts are purchasing the hardware with enthusiasm only to discover, after they buy their computer, that little educational software exists. For example, APPLE II, the most popular microcomputer for education, has only 50 hours of courseware. Most of this material is not continuous curriculum. Another problem is maintenance and repair support. Service provided for the equipment is very limited. Some schools have had to return computers to the factory for repair.

Lack of instructional software has not been a significant deterrent. Some administrators believe that their students need exposure to computers even though the courseware offerings are limited. Two commercial companies (Milliken and Foundation for Quality Education) have developed computer assisted instruction curriculums. Most schools are using their microcomputers to teach computer programming (BASIC). Elementary schools are using them to provide simulations and drill and practice instruction, mostly in math. A few schools are using microcomputers for test scoring and classroom management. Computer literacy is being emphasized, but certainly not as much as it could be.

HARDWARE

The most significant advantages of the microcomputer are its color, graphics and audio capability. The possibilities for computer based instruction are certainly enhanced when various colors and pictures can be displayed on the screen (color is not important if the machine is to be used solely for classroom management.) There is no doubt that computer assisted instruction is motivating to students; the addition of color graphics and sound make the experience even more exciting.

Dependability is one of the biggest considerations to be made in procuring hardware. If there is a breakdown, service must be readily available so that the machine can be restored in a short period of time. Currently, schools must depend on their retail dealers and not all of these provide maintenance and repair service.

The need for peripheral devices varies with the application. Some vendors provide for many additional hardware features while others do not. Printers are more likely to be used as adjuncts than as sole output devices because of their high cost and lack of sophisticated graphics.

Standardization is important because the software programs available for the various microcomputers are not transportable. This means that programs written for the TRS-80 will not operate on an APPLE II and vice versa. Many districts are purchasing micros from different vendors without realizing that their hardware assortment limits flexibility in using the machines. Peripherals can be added to most personal microcomputers, but there is very little compatability between products made by different companies.

Service requirements also vary. Relating to various service companies for maintenance of the machines can be a time consuming ordeal for the school administrator.

Memory capacity is one of the more important hardware considerations for computer assisted instruction. Random access memory (RAM) is not adequate for mass program storage. This memory limitation means that most computer assisted instruction programs require one or more disc drives, which usually cost about $500 each. Tape is less expensive, but slow and unreliable. With the decreasing cost of semiconductor RAM's, there will probably be less need for disc drives in the next two to three years. Fortunately, this development will reduce the need for saving memory space through painstaking programming methods.

The new bubble memories will dramatically increase the micros' capacity to a million or more bits. Bubbles will be much cheaper and smaller than disc memories. And, unlike RAM's, they will retain information permanently without requiring a continuous flow of electrical current.

Since distributing and sharing microcomputer software can be handled through a centralized computer system, it is important that microcomputers be able to communicate through standard modems to a host machine. A California based company, NESTAR, has conducted pioneer work in this kind of networking.[9] The compatability of various types of hardware is particularly critical to the computing consortium since networking can facilitate the sharing of programs between different types of machines. In the NESTAR system a "queen" PET microcomputer controls access to a dual eight inch floppy disc and to a printer. Several microcomputers can be added via cluster bus as long as the mixture of APPLE's, PET's or TRS-80's does not exceed fifteen.

Many districts are procuring equipment without knowing much about hardware features. These considerations seem to be the most important:

1. Does the computer include a self-contained video monitor?
2. If the machine is to be connected to a standard television set, are the appropriate interface circuits and cables provided? If so, are they approved by the Federal Communications Commission? (A major advantage of many micros is their capacity to use existing television sets as monitors.)
3. Does the computer have graphics capability?
4. Can the computer save information and programs on cassette tapes with the help of a low cost cassette recorder? If so, is the appropriate circuitry (cassette interface) included with the computer, or must it be purchased separately?
5. Are a wide range of peripherals available that will permit the computer to be upgraded after it has been purchased? Does it have sockets for additional memory and circuit cards?
6. Is the computer designed with the student in mind? Is it sufficiently durable? For example, is the keyboard legibly marked so that confusion about keys is minimized? Are lower case letters available? Is the video display terminal easy to view? If cables are required to interconnect parts of the computer, are they easy to connect and conceal?
7. Is the price of the computer and its peripherals competitive?
8. Does the manufacturer provide a warranty?
9. Is service readily available?
10. Are the manufacturer and supplier reputable?
11. Is the speed adequate?
12. Is the memory capacity adequate for lesson data and student performance data? (Most microprocessors can access up to 65K 8-bit butes of RAM memory. The typical microcomputer has between 4K and 16K bytes of RAM.)
13. Does the instruction set enable the processor to perform operations expeditiously? (The Z80 and 8085 microprocessor are generally considered superior to most other 8-bit microprocessors currently available.)[10]
14. Does the computer use a well-known higher level language like BASIC? Is the language permanently stored in a RAM memory from a cassette tape when the computer is turned on? (Higher level languages like PASCAL and PILOT are better for computer assisted instruction development.)
15. How much software is available that will run on the machine? Is it well documented?

SOFTWARE

The need for micro software is becoming more critical as the number of machines in the market place proliferates. Both system software and applications software are needed by educators.

Most microcomputers use BASIC language, which makes them particularly attractive for teaching computer programming. Manuals that teach the BASIC language accompany most micros. Many buyers are deluded in thinking that they can easily learn to write programs of their own. Actually, it takes an experienced programmer to write good computer assisted instruction and computer managed instruction programs. Courseware developed for computer based instruction takes a long time.[11] Sophisticated software requires long and detailed instructions, which means that most users will procure, rather than develop, their programs.

Unfortunately, few commercial programs exist. Buyers are usually disappointed to find out that applications software at the elementary and secondary levels is a limited commodity. Currently, there are very few commercial programs that have significant educational value.

Most educators who have bought microcomputers do not know how to find software that will operate on them. Schools that purchase Commodore PET's, for example, need access to a library of programs so that they can evaluate and then procure the ones that meet their needs.

At this time, there is a communication void in this area. Users often do not know what is available to run on their machines because most of the available software has been developed by individuals rather than commercial companies. Information about software availability is not synthesized in any single publication.

The Minnesota Educational Computing Consortium has addressed this problem of limited software by transfer mechanism, referred to as "off-loading," involves electronically moving computer programs and data files between their timeshare system and a microcomputer. Off-loading can be accomplished in two directions: "down-loading" transfers programs from the host computer to the microcomputer and "up-loading" transfers the program from the microcomputer to the timeshare system.[12]

---

[9] Technology Education Research Center (TERC) Newsletter, January 1979, p. 28.

[10] Consumer Guide, Home Computers, (Skokie, Illinois: Consumer Guide Magazine Leisure, 1978).

[11] Kent T. Kehrberg, "Microcomputer Software Development: New Strategies for a New Technology", Proceedings of the National Computing Conference (Iowa City, Iowa, June 25-27, 1979), pp. 343-346.

Down-loading provides users with an opportunity to group programs on diskettes in an arrangement which better suits their instructional needs. For example, social studies programs can be put on a single disc, while science programs can be put on another. Naturally, programs available on a time share system do not make use of the micro's color and graphics capabilities, but this facility does address the short-term need for increased accessibility of programs.

School users must have ready access to information about available courseware. Every school district does not need to conduct time-consuming research to find out about existing programs. The Northwest Regional Laboratory recently received funding to set up a software clearinghouse for educational use.

Information sources about software are extremely limited. Many programs are not well documented. Currently, there are four major sources:

1. Hardware manufacturers - Naturally, companies are interested in software that runs on their machines, but, to this date, few have undertaken developments of their own. Instead, companies like Apple and Texas Instruments have secured courseware by contract or through a royalty arrangement. Companies have accelerated their efforts (e.g. the recent affiliation of Apple with Bell & Howell) because they realize that people may not buy their equipment if there is no software to run on it.
2. Commercial Software Development Companies - There are about 100 small companies that are now developing software for the microcomputer. At a recent computing conference, some of the large publishing companies sent representatives (e.g. SRA, Random House, McGraw-Hill and MacMillan), but most of them are not making any commitments about their intentions. Scott Foresman is already developing a reading program for the Texas Instruments microcomputer. Rumors about their potential entry in this market abound. There are positive indications that these large companies will move into the field in the near future.
3. Individual School Districts and Service Agencies - Among school districts, Dallas Independent School District leads the nation in developing micro courseware. Districts in New York, Idaho, California and Minnesota have also made some contributions. The problem with much of this software is that it is often developed around local curricula.
4. User's clubs and exchanges - These groups publish newsletters which contain operating hints and tips. Documentation for software is also provided for a particular machine.

Thus far, the paucity of programs has been emphasized. However, the lack of quality is just as important. There are few good micro-based programs. Most of the computer curricula developed for the micro are "canned" versions of conventional instructional methods and materials. Too often, developers are creating micro programs which duplicate the ones already operating on minis and mainframes. In other words, the new micro technology which permits color, graphics and sound has not yet been fully exploited for use in computer assisted instruction.

FUTURE TRENDS

Despite the limitations pointed out, the computer is coming. There is no doubt that it can make a significant contribution. Several trends will promote the microcomputer's acceptance.

Education and training are gradually shifting from a teacher-centered orientation to a student-centered one. The productivity of students, rather than the productivity of teachers, is becoming the basic focus in evaluating the success and efficiency of instruction. If instruction is to improve, then student productivity must be emphasized. The use and acceptance of computers should flourish in a student-centered learning environment where individual productivity is emphasized.

Within two years, microcomputers will be within the economic reach of the average American household.[13] The availability of computer games will spark interest in the microcomputer. Children are already deluged by computer toys. Not surprisingly, today's students are more computer literate than their parents. These children will live in a much different world. According to a recent estimate, eighty percent (80%) of the jobs facing our elementary students do not even exist today. By 1985, seven out of ten adults will need to use a computer in their jobs. If this statistic is even half true, then the time to prepare students is now. Learning to write a computer program will certainly become one of the basic skills of the future.

There is little doubt that life is going to be incalculably changed by the micro technology. Isaac Asimov, a futurist quoted in Time's special issue on the computer's impact on society, summed up the issue: "We are reaching the stage where the problems that we must solve are going to become insolvable without computers. I do not fear computers. I fear the lack of them."[14]

Can education survive if educators resist the advent of microcomputers? Probably not, according to Dr. Seymour Papert of MIT, who recently gave this testimony to Congress:

> During the 1980's small, but intensely powerful computers will become as much a part of everyone's life as the TV, telephone, the printed paper and the notebook. Indeed, computers will integrate and supersede the functions of

---

[12] Microcomputer Report of the Minnesota Educational Computing Consortium, (Lauderdale, Minnesota: MECC Instructional Services Division, 1979), p. 19.

[13] "Living: Pushbutton Power," Time, February 20, 1978, p. 48.

[14] "The Age of Miracle Chips", Time, February 20, 1978, p. 44.

> these and other communicational and recreational home technologies. I emphasize: this will happen independently of any decisions by the education community; the driving force lies in industry. This computer presence has a tremendous potential for psychological impact including improvement of the process of learning. It will affect adults, children and babies in homes, as well as in schools .... Every child will have access to computing power ....[15]

---

[15] Computers and the Learning Society: Report. Subcommittee on Domestic and International Scientific Planning, Analysis, and Cooperation of the Committee on Science and Technology. U.S. House of Representatives, Ninety-fifth Congress, Second Session, June 1978, p. 32.

THE USAGE OF SHORT INTERNAL SORT ALGORITHMS FOR TEACHING MULTIPLE APPROACHES TO ELEMENTARY COMPUTER FUNCTIONS

Jerry D. Whitt, Millsaps College
C. Nathan Adams, Middle Tennessee State University

ABSTRACT

Seven short internal sort algorithms are presented in the Basic language. The algorithms are useful as a pedogogical device for underscoring the point that there are multiple valid approaches to a computer solution for a particular problem. As a secondary benefit the problem happens to be a basic function which all information systems students must learn: sorting lists into ascending and/or descending sequence.

The seven algorithms were selected because of their brevity, and cover all four of the comparative sort categories. Relative efficiencies of the seven algorithms are also presented.

INTRODUCTION

As a pedogogical device for teaching a fundamental data processing function and introducing some basic concepts about algorithm efficiency, seven short internal sort routines[1] were presented to a sophomore-level data structures class. The students were assigned a Basic programming project of sorting 50 records[2] using whichever algorithm interested them and measure the sorting time. In addition to learning a fundamental function, internal sorting, two concepts were introduced which are essential for beginning information systems majors to understand but which are not intuitively apparent: algorithm efficiency and algorithm sensitivity to data characteristics. To provide consistency for the comparative aspects of the project, the seven algorithms were also run in a more controlled environment.

The purpose of this paper is two-fold. First, the results of the comparative study will be presented. Second, the seven algorithms will be presented. The second part will be tuitorial in nature and will be beneficial to anyone who wishes to use the algorithms as a teaching technique.

COMPARATIVE STUDY OF SORT ALGORITHMS

The programs for the comparative study were run with 50 key values (hereafter referred to as the key set) generated using the random number generator and lapse time was measured by the internal clock. Since there can be some time variability from run to run in Basic, even with the same program and data, runs were made with three different sets of random numbers and the times reported in Table I are averages of these three runs. The deviation from run to run was trivial.

The concept of algorithm efficiency is made apparent by a row-by-row comparison in Table I. Disregarding the variance among columns (which represents key set variability), that some algorithms are more efficient than others is easily discernible.

The second concept which was introduced by the study--algorithm sensitivity to data characteristics--can be seen by making a column-by-column comparison of Table I. Since the efficiency of some algorithms is dependent upon the distribution of the key set, each algorithm was run also on three distributions of the same key set values. The random numbers were sorted, the sorted key set was finally resorted into reverse order. Thus, each algorithm could be studied on a key set already in perfect sequence, a key set in reverse of the sequence desired and a key set distributed somewhere between those two extremes.

The data of Table I represents sorting of only the key set. Some algorithms are sensitive to record length, which would consist of the key set and associated data fields, because they involve considerable data movement. In the programs in the Appendix, whenever the X array is used in a LET statement a key is being moved and a series of moves (one for each field) would have to be inserted at that point.

INTERNAL SORT ALGORITHM TUTORIAL

General purpose internal sort algorithms can be classified as one of four types according to Knuth [1,73].

---

[1] Selected from [2 and 3]. The basis for selection was the program's length: no program requiring more than a dozen (an arbitrary number) instructions, excluding input/output, was included. All seven fall under the category of comparative, as opposed to distributive, algorithms. The seven Basic programs are presented in the Appendix.

[2] Each record had a key and five fields.

223

TABLE I

COMPUTER SORT TIME* FOR A KEY SET OF 50 OCCURRENCES

SORT ALGORITHM	RANDOM KEY SET	KEY SET IN SEQUENCE	KEY SET IN REVERSE SEQUENCE
INSERTION SORTS:			
Linear Insertion with Exchange	.28	.01	.54
Ranking Sort	.16	.02	.30
Shellsort	.11	.06	.14
EXCHANGE SORT:			
Standard Exchange	.37	.20	.53
SELECTION SORTS:			
Linear Selection	.27	.26	.33
Successive Minima	.21	.20	.21
ENUMERATION SORT:			
Enumeration	.37	.37	.37

*Times are averages of three computer runs and are rounded to the nearest hundredth of a second.

1. Insertion sort. The items are considered one at a time, and each new item is inserted into the appropriate position relative to the previously-sorted items. (This is the way many bridge players sort their hands, picking up one card at a time.)

2. Exchange sort. If two items are found to be out of order, they are interchanged. This process is repeated until no more exchanges are necessary.

3. Selection sort. First the smallest (or perhaps the largest) item is located, and it is somehow separated from the rest; then the next smallest (or next largest) is selected, and so on.

4. Enumeration sort. Each item is compared with each of the others; counting the number of smaller keys determines the item's final position.

INSERTION ALGORITHMS

The most efficient insertion algorithms utilize a technique called sifting, which involves two looping processes. The outer loop goes down the list and as each new key value is considered in the list, an inner loop compares and pushes the value up the list to its proper position. As the outer loop increases, in other words, sifting algorithms compare value_i with preceding values ($\text{value}_1 \ldots \text{value}_{i-1}$), which already have been sorted, and inserts value_i in its proper position. Three different sifting algorithms are considered: linear insertion with exchange, the ranking sort, and the shellsort.

Exhibit 1 illustrates linear insertion with exchange. Assuming the outer loop has been executed three times, so the first three values are already in sequence, the fourth value is compared and exchanged up the list until it is greater than any predecessors at which point the inner loop would be completed for that key and control would revert to the outer loop.

The ranking sort is similar in concept except the key value in the ith position (the 4th position in Exhibit 2), would be set aside and used for comparison with each predecessor in the list. Instead of exchanging out of sequence key values, however, values greater than the one set aside each would be pushed down until the correct position was found. The value which was set aside would then be inserted into that position.

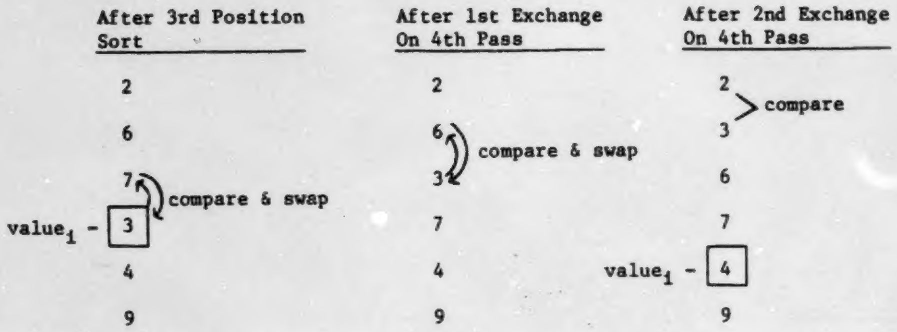
The third insertion sort routine considered, the Shellsort, is the fastest of all seven algorithms if the record consists only of the key field. The theory underlying the Shellsort is that if an exchange is necessary on early passes through the key set, values of considerable distance apart should be swapped, not just values in adjacent positions. Thus, the file is initially divided into two partitions with the 1st and 26th (if there are 50 key values) values compared and exchanged if necessary, the 2nd and 27th, and so on. The key set is then divided into four partitions with the 1st key value being compared to the 13th, and 2nd with the 14th, and so on in the first partition, and the 26th with the 38th, the 27th with the 39th and so on in the second partition. The number of partitions are doubled after each pass until there are as many partitions as there are key values. The last pass through the data then is identical in concept to linear insertion with exchange, with each key value being pushed up the list to its appropriate position. Only eight key values are used to illustrate the Shellsort in Exhibit 3, so the first and fifth key values, second and sixth key values, and so on, are compared and swapped if necessary on the first pass.

EXCHANGE ALGORITHMS

As has been seen, several algorithms incorporate exchange concepts. The standard exchange, or bubble sort, is slow but easy to understand and code. It, along with the ranking sort, is the shortest routine presented.

EXHIBIT 1

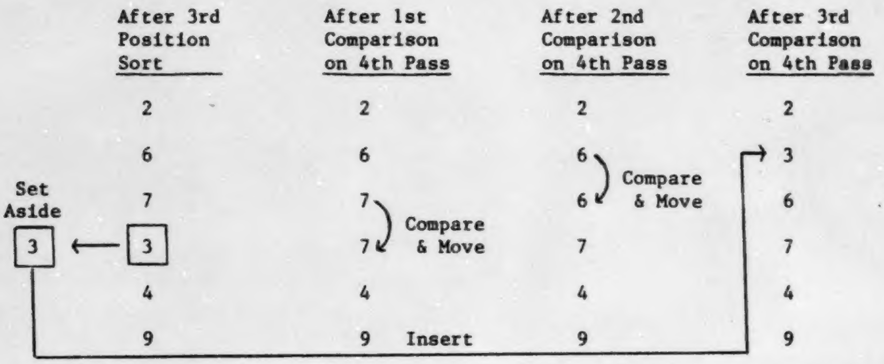
LINEAR INSERTION WITH EXCHANGE


After 3rd Position Sort
2
6
7
value$_1$ - 3
4
9
compare & swap
After 1st Exchange On 4th Pass
2
6
3
7
4
9
compare & swap
After 2nd Exchange On 4th Pass
2
3
6
7
value$_1$ - 4
9
compare

As Table I indicates, the efficiency of linear insertion with exchange is more dependent upon the key set distribution than any of the other algorithms studied.

EXHIBIT 2

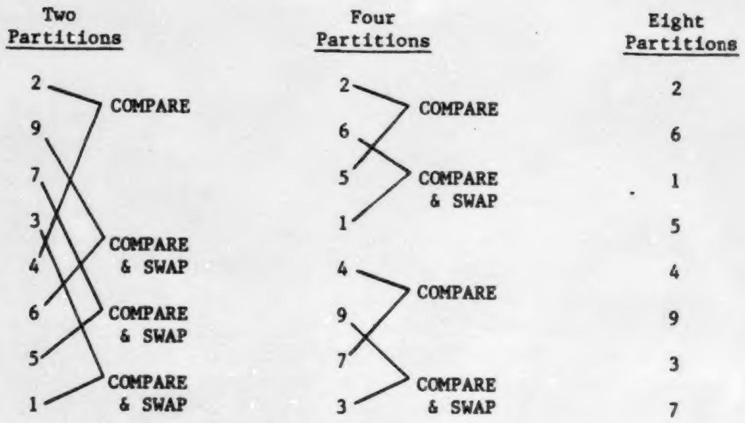
RANKING SORT*


After 3rd Position Sort
2
6
7
3
4
9
Set Aside
3
After 1st Comparison on 4th Pass
2
6
7
7
4
9
Compare & Move
Insert
After 2nd Comparison on 4th Pass
2
6
6
7
4
9
Compare & Move
After 3rd Comparison on 4th Pass
2
3
6
7
4
9

*Assumes the first three positions have already been sorted and the text describes the process beginning with the fourth position.

EXHIBIT 3

SHELLSORT


Two Partitions
2
9
7
3
4
6
5
1
COMPARE
COMPARE & SWAP
COMPARE & SWAP
COMPARE & SWAP
Four Partitions
2
6
5
1
4
9
7
3
COMPARE
COMPARE & SWAP
COMPARE
COMPARE & SWAP
Eight Partitions
2
6
1
5
4
9
3
7

Using the bubble sort, repeated passes are made through the key set with adjacent key values swapped if necessary. In effect, the largest value of the key set is moved down to the bottom of the list and the loop length becomes shorter by one key value with each iterative process. The bubble sort is really a selection-exchange process.

SELECTION ALGORITHMS

Two selection algorithms are presented and both are efficient when sorting records containing numerous fields.

The linear selection process locates the smallest value in the list and copies it to the next available position in a second list. A high, nonvalid value is then inserted into that location of the original list and the process is repeated.

The successive minima routine searches the key set to find the smallest value and exchanges it with the key set in the first position. Beginning with the second position, the search continues for the smallest remaining value which is exchanged with the key in the second position, and so on.

ENUMERATION SORT

Enumeration is not a minimal storage technique. In addition to the array for the keys, arrays are required for the counters and as a recipient area for the copying. A counter is established for each key value when using an enumeration sort. Each value is compared with all other values and one is added to the counter of the larger comparand. After all combinations have been compared, the counter value plus one will represent the key value's relative position in the set and each value can be copied into that position in a second array.

SUMMARY

Rather than having been assigned a "best" internal sort algorithm, information systems majors were introduced to seven alternative approaches and were assigned a programming project which would utilize whichever algorithm most interested them. The results of a comparative study was then presented which introduced them to the concepts of algorithm efficiency and algorithm sensitivity to data characteristics.

APPENDIX

Basic Sort Algorithms

The key set in all of the programs is located in array X and is copied to array Y in the non-minimal algorithms. Array C is used for the counters in enumeration. I is always the index of the outer loops and J the index for the inner loops. T is used as a temporary holding field for those algorithms requiring one, M is the partition range in the Shellsort, and K is a holding field for an index in linear selection. The programs are set up for 50 key values.

Linear Insertion with Exchange

```
 10  DIM X(50)
 20  FOR I = 1 TO 49
 30  LET J = I
 40  IF X(J + 1)> = X(J) THEN 100
 50  LET T = X(J)
 60  LET X(J) = X(J + 1)
 70  LET X(J + 1) = T
 80  LET J = J - 1
 90  IF J> = 1 THEN 40
100  NEXT I
```

Ranking Sort

```
 10  DIM X (50)
 20  FOR I = 1 TO 49
 30  LET T = X(I + 1)
 40  FOR J = I TO 0 STEP - 1
 50  IF X(J)< T THEN 80
 60  LET  X(J + 1) = X(J)
 70  NEXT J
 80  LET X(J + 1) = T
 90  NEXT I
```

Shellsort

```
 10  DIM X(50)
 20  LET M = 50
 30  LET M = INT (M/2)
 40  FOR I = 1 TO 50 - M
 50  LET T = X(I + M)
 60  FOR J = I TO 0 STEP - M
 70  IF X(J)< T THEN 100
 80  LET (J + M) = X(J)
 90  NEXT J
100  LET X(J + M) = T
110  NEXT I
120  IF M> 1 THEN 30
```

Standard Exchange

```
 10  DIM X(50)
 20  FOR I = 49 TO 1 STEP - 1
 30  FOR J = 1 TO I
 40  IF X(J)< = X(J + 1) THEN 80
 50  LET T = X(J)
 60  LET X(J) = X(J + 1)
 70  LET X(J + 1) = T
 80  NEXT J
 90  NEXT I
```

Linear Selection

```
 10  DIM X(50), Y(50)
 20  FOR I = 1 TO 50
 30  LET T = X(1)
 40  LET K = 1
 50  FOR J = 2 TO 50
 60  IF X(J)> = S THEN 90
 70  LET T = X(J)
 80  LET K = J
 90  NEXT J
100  LET Y(I) = T
110  LET X(K) = 9999
120  NEXT I
```

Successive Minima

```
 10  DIM X(50)
 20  FOR I = 1 TO 49
 30  LET K = I
 40  FOR J = I + 1 TO 50
 50  IF X(J)> = X(K) THEN 70
 60  LET K = J
 70  NEXT J
 80  IF K< = I THEN 120
 90  LET T = X(I)
100  LET X(K) = X(K)
110  LET X(K) = T
120  NEXT I
```

Enumeration

```
 10  DIM X(50), C(50), Y(50)
 20  FOR I = 1 TO 49
 30  FOR J = I + 1 TO 50
 40  IF X(I)< = X(J) THEN 70
 50  LET C(I) = C(I) + 1
 60  GO TO 80
 70  LET C(J) = C(J) + 1
 80  NEXT J
 90  LET Y(C(I) + 1) = X(I)
100  NEXT I
110  LET Y(C(50) = 1) = X(50)
```

REFERENCES

1. Knuth, Donald K., The Art of Computer Programming, Volume 3/Sorting and Searching, Addison-Wesley Publishing Company, Reading, MA, 1975.

2. Lorin, Harold, Sorting and Sort Systems, Addison-Wesley Publishing Company, Reading, MA, 1975.

3. Rich, Robert P., Internal Sorting Methods Illustrated with PL/1 Programs, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1972.